Social Science Computer Review
2024, Vol. 42(3) 760–776
© The Author(s) 2023
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/08944393231215459
journals.sagepub.com/home/ssc
Sage

Bridging Qualitative Data Silos: The Potential of Reusing Codings Through Machine Learning Based Cross-Study Code Linking

Sergej Wildemann[1], Claudia Niederée[1], and Erick Elejalde[1]

Abstract

For qualitative data analysis (QDA), researchers assign codes to text segments to arrange the information into topics or concepts. These annotations facilitate information retrieval and the identification of emerging patterns in unstructured data. However, this metadata is typically not published or reused after the research. Subsequent studies with similar research questions require a new definition of codes and do not benefit from other analysts' experience. Machine learning (ML) based classification seeded with such data remains a challenging task due to the ambiguity of code definitions and the inherent subjectivity of the exercise. Previous attempts to support QDA using ML rely on linear models and only examined individual datasets that were either smaller or coded specifically for this purpose. However, we show that modern approaches effectively capture at least part of the codes' semantics and may generalize to multiple studies. We analyze the performance of multiple classifiers across three large real-world datasets. Furthermore, we propose an ML-based approach to identify semantic relations of codes in different studies to show thematic faceting, enhance retrieval of related content, or bootstrap the coding process. These are encouraging results that suggest how analysts might benefit from prior interpretation efforts, potentially yielding new insights into qualitative data.

Keywords

qualitative coding, qualitative data, machine learning, computational social science

Introduction

Qualitative data analysis (QDA) is one of the primary research approaches in social science, psychology and other human sciences, alongside quantitative analysis. For QDA, analysts

[1] L3S Research Center, Germany

Corresponding Author:
Sergej Wildemann, L3S Research Center, Leibniz Universität Hannover, Appelstr. 9A, Hannover 30167, Germany.
Email: wildemann@l3s.de

typically use unstructured textual data from sources such as interviews, questionnaires, or field notes (Flick, 2013). In a process called "coding," researchers annotate snippets of this unstructured textual data with a set of codes (a kind of controlled labels) representing aspects or concepts of interest for the particular study (Bernard, 2013). Codes are used to categorize and organize the data, making it easier to identify patterns and relationships. The set of codes can be theory- and/or data-driven and is refined iteratively as the researchers go through the textual material. Researchers then explore the relationships between themes, codes, and data segments. They examine connections, patterns, and variations within the data to develop a comprehensive understanding of the research topic.

Today, it is easier than ever to collect, share, and re-use rich information on many aspects of society. With the trend towards more open and reproducible research, valuable datasets are becoming increasingly available (see, for example, the UK Data Archive (University of Essex, 2023)). On the other hand, the availability of ever-larger datasets exacerbates the QDA challenges of coding and managing information. For example, each research question often prompts an adapted codebook (coding) without much benefit from prior experience, either one's own or that of others. At the same time, high-quality coding still requires expert knowledge and is therefore usually expensive and time-consuming. Thus, the use of intelligent methods to explore and retrieve shared annotated snippets relevant to the adapted codebook would greatly improve the efficiency of QDA (Jiang et al., 2021).

Recent developments in natural language processing (NLP) and language modeling have expanded the ability of machine learning (ML) techniques to capture contextual and semantic information (Paredes et al., 2017; Young et al., 2018). These technologies improve text classification tasks, including the autocoding of qualitative data (Kaufmann et al., 2020). In addition, accurate language-based classifiers open up promising opportunities to link semantically similar bits of data across studies, helping researchers to find and reuse relevant information for their computer-assisted QDA (CAQDA) (Hienert et al., 2019).

The creation of codes is difficult to automate, as it depends heavily on the research questions and the analysts' experience and interpretation of the text (Chen et al., 2018). Also, qualitative scientists favor a more human-involved process over automatically generated codes (Marathe & Toyama, 2018). However, the extension of defined codes to unseen documents for discovering and recommending related content could benefit from ML-based tool assistance (Jiang et al., 2021).

In this study, we explore steps towards linking concepts from multiple studies by finding semantic relationships of cross-dataset codes. This can provide analysts with further thematic faceting, enhance retrieval of related content, or bootstrap the coding process. We also evaluate the performance and generalizability of several classifiers on three relatively large, manually annotated, real-world datasets from qualitative studies. All models are trained and evaluated on analyst-defined coding schemes and a subset of manually annotated samples.

Background and Related Work

Coding of Qualitative Data

Qualitative research uses a variety of analytical methods to gain insights from data. These are characterized by inductive steps to organize datasets and find interesting patterns and themes and deductive steps to validate existing theories on the data. Grounded theory (Glaser et al., 1968; Strauss & Corbin, 1997) thereby offers a set of methods in which data collection, analysis, and interpretation alternate to generate theories that are "grounded" in the data. Following on from this, the *Grounded Theory Method* (Charmaz, 2006) provides a specified set of practices for the coding process and the approach to data. In an open coding stage, data is exploratively labeled and categorized. Relations between codes define broader categories, and the constant comparison of data allows for the discovery of differences and the development of initial theories. The emerging theories guide the further collection process and are tested for weaknesses. After iterative refinement, the set of codes and categories has become more abstract and grounded. A subset of the inductively generated coding scheme then forms the focus of the study and is selected to be deductively applied to the entire dataset for deeper analysis.

Researchers commonly resort to computer-assisted QDA software (CAQDAS) to streamline the process of organizing, coding, and analyzing data, making it more efficient and accessible. MAXQDA,[1] Atlas.ti,[2] and NVivo[3] are popular examples among social scientists (Freitas et al., 2017). These tools help researchers manage large volumes of unstructured data, assign codes, and identify patterns and themes, thus improving the rigor and consistency of qualitative research. In addition, CAQDAS enables collaboration, allowing multiple analysts to work simultaneously on the same dataset. However, the main focus of most CAQDASs is still on supporting manual coding and the subsequent analysis (Marathe & Toyama, 2018), with limited assistance for autocoding (i.e., applying the codebook to unseen documents) and code augmentation. Adding AI-based functionality to CAQDAS that utilizes new ML developments would be an advantageous contribution.

Supporting the Coding Process via Machine Learning

There have been attempts to automate various stages of the coding process. For the open coding stage, semantic similarities (Paredes et al., 2017), and approaches like graph theory (Tierney, 2012), or topic modeling (Bakharia et al., 2016; Baumer et al., 2017) were used to extract latent relations in the data and create new categories. These methods integrate computer-aided exploration of the data into the inductive process.

ML can support collaborative coding at early stages by identifying and highlighting ambiguity. The coding exercise is inherently subjective (Knoblauch, 2013), and discrepancies between coders may indicate ambiguity in the code definitions or specific data points that collaborators need to discuss. Besides, these elements may reveal complexities in the dataset that could trigger the need for new codes and/or reconsideration of already labeled data (Chen et al., 2018). Instead of focusing on recommending codes, Drouhard et al. presented *Aeonium*, a system that leverages interactive ML to identify ambiguity during coding (Drouhard et al., 2017). It trains a support vector machine (SVM)-based classifier for each user to predict segments where coding partners may disagree, promoting consistency and mutual understanding.

Further, automatic text classification can help extend the coding to unseen snippets. Several contributions were made on this subject, while two main approaches stand out: rule-based (Crowston et al., 2010, 2012) and ML-based (Crowston et al., 2010; Drouhard et al., 2017; Liew et al., 2014). Crowston et al. contrasted a linear classifier operating on different feature sets with a rule-based approach, with the latter performing slightly better, especially in the case of scarce training examples (Crowston et al., 2010). In particular, the problem of identifying a good feature set was highlighted. For this purpose, a newly created dataset from a discussion forum for developers was coded. Crowston et al. then followed up on this work and deepened the exploration of custom NLP rules to identify text segments of interest and provide an initial coding (Crowston et al., 2012). The authors found problems with unit coding, sparse codes, and manual coding errors in the training data. Furthermore, they note that NLP would be unlikely to work with codes that are highly dependent on subjective interpretation and context. As the development and validation of such rules required considerable additional effort and experience, it was not considered practical for smaller datasets. Nevertheless, it was seen as a way to enable elaboration on larger datasets. Yan et al. used a modification of the same dataset to investigate the use of machine learning to support initial coding (Liew et al., 2014). Instead of manually searching for suitable text segments, a SVM-based model should be trained iteratively during coding in an active learning process and provide suggestions to the analyst. The results were therefore geared towards high recall and achieved only low precision, even with the best codes. Further problems with other codes were the small number of examples and the complexity of the theoretical concepts to be captured.

Keyword or query-style matching rules are challenging and require more analyst involvement, but their results are easier to explain. ML is more generalizable and easy to apply for the end user, but the results may not be directly apparent to researchers. A combination of both methods has also been proposed via user-defined query-style code rules merged with supervised ML (Rietz & Maedche, 2021). Rietz and Maedche proposed a user-centric methodology based on Interactive Machine Learning (IML) for a more transparent learning process. Supporting the coding process with rule and ML-based suggestions resulted in higher intercoder reliability compared to coding the same interviews using other CAQDAS. However, they found that the amount of training data available limited the quality of their ML-based code suggestions. In addition, the consistency and error-free nature of the coded data plays an important role in the training of a model. Given sufficient budget, double coding, that is, coding the same text segments by several people, can reveal difficult or ambiguous parts and thereby improve the training data (He & Schonlau, 2020).

Similarly, we want to explore the potential of ML during the coding phase and beyond. Still, in our task, unlike previous autocoding attempts, the content to be classified does not come from the same study, and the underlying concepts may overlap to various extents. To this end, we base our investigation on relatively large datasets from three social science studies, whereas previous approaches have used either small or custom-built datasets.

Transfer Learning in Cross-Domain Text Classification

Most previous ML-based approaches to autocoding used more classical methods such as SVM and logistic regression. Meanwhile, the state of the art in many NLP tasks is dominated by pre-trained encoders (Tenney et al., 2019) that produce a vector representation of the textual input that can be used, for example, for downstream classification. Particularly, transformer-based language models like BERT (Devlin et al., 2019) achieve top performance on standard NLP benchmarks (Wang et al., 2019). These models help to overcome the limitations of classical neural networks. The most evident is that they allow transfer learning across domains, alleviating the need for extensive training datasets. This is especially relevant in contexts such as qualitative coding, where datasets are often comparatively small and specific. Also, bidirectional language models like BERT or ELMo (Peters et al., 2018) learn deep contextualized word embeddings that capture complex characteristics (e.g., syntax and semantics) and their variation across linguistic contexts. Our main problem (i.e., cross-dataset label linking and codebook relationship analysis) is expected to benefit from such improved text classification methods. So, we build on previous studies by incorporating new state-of-the-art language modeling into the pipeline.

Finally, our work also draws from previous research in transfer learning. In ML, transfer learning applies a pre-trained model to new problems. It entails reusing the knowledge and insights learned from a prior task to enhance the model's ability to generalize to a different task (or a similar task in a different domain) (Weiss et al., 2016). Typical applications of transfer learning include training in a resource-rich domain to boost performance in another domain where annotated data points are scarce (for example, diagnosis of rare diseases (Taroni et al., 2019), in cross-country/cross-language analysis (Wildemann et al., 2023), or qualitative environmental research using spectral data (Cui et al., 2019). In this context, feature-based transformation strategies are frequently employed. In cases like ours, we have a cross-domain text classification challenge, where the objective is to create a target classifier using labeled text data from a related domain. In such a scenario, a viable approach is to identify common latent features, such as latent topics (Zhuang et al., 2020).

In the same vein, some studies on cross-domain recommendation also extract semantic relationships across datasets for content recommendation. However, most of the work in this area differs from ours in that it either focuses on unidirectional knowledge transfer (Zhao et al., 2020), includes extra information about user preferences (Xie et al., 2021), or relies on case-specific features (e.g., the reference network in scientific publications) (Ebesu & Fang, 2017; Xie et al., 2021). For linking qualitative codes, we only have the text snippets for each label as input. Furthermore, our texts tend to convey a more general language (rather than scientific, concise, or curated), and the annotations are more subjective and tailored to each dataset.

Datasets

For our experiments, we rely on three datasets from German qualitative studies in the social sciences, specifically in the sociology of work. These empirical surveys consist of group discussions and individual interviews that were manually coded according to study-specific codebooks. Multiple researchers were involved in the coding process, while a single person coded each document. For our goal of finding relationships between codebooks from different studies, these datasets are particularly suitable as they address different questions in the same research field, but were created by distinct research groups with varying coding styles. The datasets were provided by the respective institutes as part of a research project and cannot be published due to confidentiality requirements. However, we summarize their main characteristics in Table 1 and describe them in detail below.

- The *BrueLeg* set consists of about half of the study material from a research project on the demands and interest orientations of employees against the background of experiences with work crises such as cost reduction or restructuring (Kratzer et al., 2019). It represents our largest dataset, with 6446 separate text samples spanning 153 interviews.
- *ZuL* extends over two case studies from a research project investigating the causes and forms of time and performance pressure, how they are perceived, and how employees and managers cope with them (Dunkel & Kratzer, 2016). Unlike BrueLeg, the codes in this dataset are organized in a two-level hierarchy to group semantically similar topics, which will prove helpful in our experiments.
- *Bank* is part of the interviews conducted in a study to address operational measures and individual competencies that lead to a better work-life balance for employees (Kratzer et al., 2015). The specific subset revolves around the banking sector and is the smallest dataset in

Table 1. Summary of Datasets in Our Experiments. The Last Two Studies Organized Their Codes in a 2-Level Hierarchy.

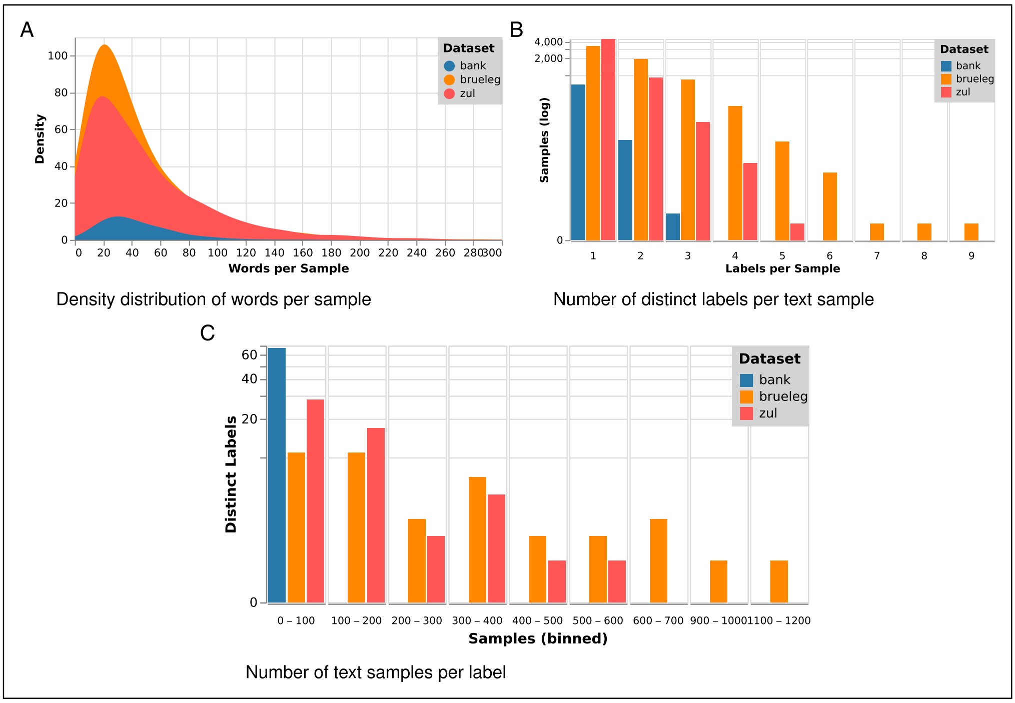
		Codes			
Dataset	Interviews	*Top-Level*	*2nd-Level*	Text Samples	Code assignments
BrueLeg	153	–	41	6446	11175
ZuL	50	9	54	5511	6766
Bank	16	13	67	727	795

our collection, but it also has the largest number of codes. As with ZuL, *Bank*'s codebook is organized into two levels.

The rule for coding was to select thematically self-contained passages (samples) to provide sufficient context for later analysis of the set's codes. The length of the samples varies from a few sentences to entire paragraphs (see Figure 1(a)). A majority of the samples contain multiple sentences from the interviewee's response, while the corresponding question is included only when necessary (e.g., affirmative answer). The average number of words per preprocessed sample (see Sec. Methodology) is around 45 for all datasets. The choice of codes (or labels) depends largely on the type of interview and the dominant topics in the conversation, resulting in an unbalanced distribution. Each pairing of a text sample with a label is counted as an *assignment* (see last column in Table 1). Figure 1(c) shows that particular labels are rarely used, whereas rather broad labels occur for many samples. Moreover, multiple different labels can apply to the same text sample (see Figure 1(b)). This is the case for 48.2% of the codings in BrueLeg, 19.3% in ZuL, and 9.1% in Bank.

Ethical Considerations

While our current research does not directly involve human participants or report on individual data, the datasets used in our study are derived from previous surveys that did implicate individuals, their personal data and inner workings of companies. We recognize the importance of ethical considerations in the broader context of social science research and want to emphasize our commitment to ethical research practices. The original surveys from which these datasets were



Density distribution of words per sample

Number of distinct labels per text sample

Number of text samples per label

Figure 1. Overview of sample and label statistics for our three datasets.

obtained were conducted in accordance with all relevant ethical guidelines and obtained informed consent from participants. The data used in our study has been de-identified and anonymized to ensure the privacy and confidentiality of the participants and their associated companies. Additionally, we have obtained the necessary permissions and adhered to any confidentiality requirements imposed by the institutions that provided these datasets. Moreover, we do not share or publish any personally identifiable information or sensitive data, in accordance with the requirements of the data providers. This includes the complete codebooks of the individual studies, which we therefore present only in part and paraphrased where necessary. Finally, our research focuses on methodological and analytical aspects rather than the content of the original participant data, and we approach this work with the utmost care for ethical considerations and data privacy.

Methodology

Our goal is to identify relations between codebooks of different studies by using classifiers trained on coded interview sections. The exported datasets consist of text segments (samples) with associated labels. Since each text sample can have more than one label, the task can be defined as a multi-label classification problem. Existing methods for multi-label classification fall into two categories: problem transformation and algorithm adaptation (Tsoumakas & Katakis, 2007). While the former transforms the problem into multiple binary classifications or regression problems, the latter directly adapts algorithms to deal with multi-label data. We selected the best performing algorithms from each category. Specifically, we experiment with Linear SVC as a one-vs-rest classifier and ML-KNN (Zhang & Zhou, 2007). Besides these established algorithms, we also include fastText (Joulin et al., 2017) and the deep learning-based model BERT (Devlin et al., 2019) as leading-edge approaches in both categories.

In order to train the classifiers, the raw text must be preprocessed and vectorized. Our preprocessing step consists of tokenization, lowercasing, and removal of stop words and numbers combined with lemmatization. To account for the exact transcription of the spoken words during the interviews and the subsequent frequency of speech disfluencies, common stop words had to be extended with fillers (e.g., "uh," "erm," "hmm"). For vectorization, we apply the term frequency-inverse document frequency (TF-IDF), as it is commonly used with Linear SVC and ML-KNN. Based on domain-specific material, including the interviews and other related studies, we further train custom word embeddings using fastText. These should capture the linguistic characteristics of the interviews and can be used instead of TF-IDF vectors when transformed into sentence vectors. Moreover, fastText can perform multi-label classification while benefiting from a pre-trained embedding model, as we will show in our experiments. Finally, BERT can work on the raw text directly. It relies on bidirectional language models pre-trained on extensive datasets. We select *gbert-base* (Chan et al., 2020) for our experiments, as this is one of the most popular monolingual German models currently available. Here, the only text cleanup necessary is the removal of special markers used in interview transcripts to separate questions and answers.

We perform hyperparameter tuning with 5-fold (80% training - 20% validation) cross-validation and stratified sampling to optimize the ML models and prevent overfitting to the training data. The target metric depends on the respective task. If used for suggestions during coding, the focus could be more on recall. Whereas if the classifier is used to identify sections in other documents that match existing codes, both high precision (P) and recall (R) are crucial. As this fits our goal, we tune the models for a high micro-averaged F_1-score. Additionally, both fastText and BERT provide a confidence score for each label. Applying a confidence threshold allows limiting the number of outputs to trade higher precision for lower recall or vice versa.

Finally, we examine the semantic similarities of the labels from the two larger datasets (i.e., BrueLeg and ZuL). For this, we train separate BERT classifiers on each dataset as before but predict labels on the other dataset. The share of samples with a label X in the test dataset that are predicted with label Y from the training dataset indicates the degree of correspondence between both labels of different codebooks. Subsequently, both cross-classification results are superimposed to identify bidirectional correlations (i.e., relations between pairs of labels from different studies on which both classifiers agreed). This approach allows us to limit the influence of uncertainties of a single classifier and to avoid excessive matches of frequent labels. Thus, by multiplying these values from both directions, we obtain a confidence score per label relation (see Figure 2). Formally, we define the relation score between labels $l_i \in D_1$ and $l_j \in D_2$ from the datasets D_1 and D_2 (*BrueLeg* and *ZuL* in our experiments) as follows:

$$lrcs\left(l_i, l_j\right) = \frac{\left\|\left\{\varkappa_a \middle| \varkappa_a \in D_2 \bigwedge f(\varkappa_a) = l_i\right\}\right\|}{\left\|\chi_{l_j}^{D_2}\right\|} \otimes \frac{\left\|\left\{\varkappa_b \middle| \varkappa_b \in D_1 \bigwedge g(\varkappa_b) = l_j\right\}\right\|}{\left\|\chi_{l_i}^{D_1}\right\|} \tag{1}$$

where $f(\blacksquare)$ is the classifier trained on D_1, $g(\blacksquare)$ is the classifier trained on D_2, $\chi_{l_i}^{D_1} \subseteq D_1$ are the snippets annotated with l_i, and $\chi_{l_j}^{D_2} \subseteq D_2$ are the snippets annotated with l_j.

Experiments and Results

This section first shows the performance of different classifiers on qualitative data and highlights some challenges. We then explore the detection of cross-study codebook relationships, demonstrate the effectiveness of our method, and discuss its potential utility for CAQDA.

Classifier Performance

The results of all models after optimization are presented in Table 2. For fastText, we include the results with and without pre-training to emphasize the improvement of using custom pre-trained word embeddings over the baseline. Since the labels in the *ZuL* and *Bank* datasets are organized in two-level hierarchies based on topics, we also report results for predicting first-level labels (suffixed with *"Base"*).

Overall, BERT achieves the best performance on the F_1 measure in all datasets except for *Bank*. Even after an extended hyperparameter search, we were not able to improve the results for this last dataset. We believe that the poor performance for *Bank* may be due to a combination of a low sample count (727) and a relatively high number of labels (67), which likewise affects the other

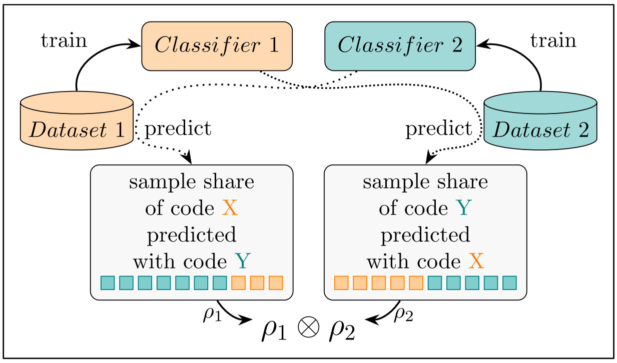


Figure 2. Cross-dataset code correlation using two classifiers.

Table 2. Micro-Averaged Multi-Label Classification Performance (%) on Coded Qualitative Research Data. FastText is Included With (PT) and Without Pre-trained Embeddings.

	BrueLeg			ZuL			ZuL Base			Bank			Bank Base		
Classifier	*P*	*R*	F_1	*P*	*R*	F_1	*P*	*R*	F_1	*P*	*R*	F_1	*P*	*R*	F_1
BERT	47.3	52.3	**49.7**	31.7	38.9	**34.9**	51.1	59.6	**55.0**	3.5	36.8	6.3	45.5	53.7	**49.2**
fastText PT	42.4	52.9	47.1	24.3	35.3	28.8	42.5	67.6	52.2	25.6	21.6	**23.3**	36.8	53.8	43.7
fastText	38.7	44.0	41.2	22.1	28.1	24.7	41.9	59.5	49.2	23.1	21.2	22.1	35.9	29.9	32.6
Linear SVC	38.1	56.4	45.5	20.2	40.2	26.9	43.4	59.6	50.3	22.5	14.0	17.2	37.1	40.6	38.8
ML-KNN	57.1	23.3	33.1	34.4	6.7	11.3	57.0	30.2	39.5	31.8	7.1	11.5	57.3	22.0	31.7

classifiers. Previous studies have also observed decreased performance in BERT fine-tuning when dealing with a small training set (Pruksachatkun et al., 2020; Zhu et al., 2020).

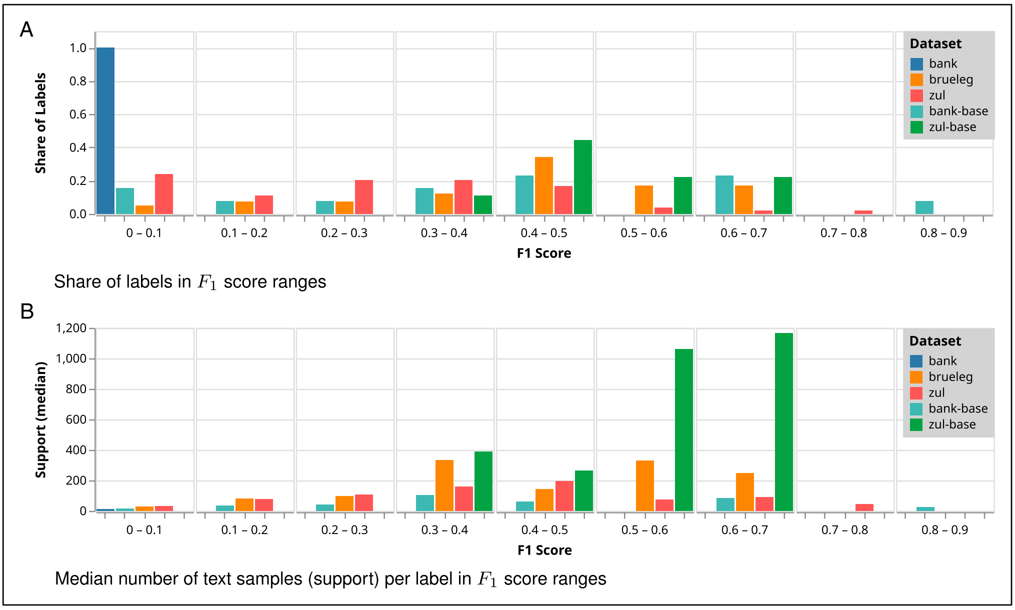
fastText proves to be the second-best classifier and even outperforms BERT on the *Bank* dataset. The additional use of pre-trained word embeddings clearly benefits the task and shows an average increase in F_1-score of 15.3% across all datasets. This suggests a further improvement when using larger domain-specific corpora. Moreover, its fast training and the lack of dependency on GPUs may be advantageous for many practical applications.

In contrast to the improvements from custom word embeddings in fastText, Linear SVC, and ML-KNN show the best results when operating on TF-IDF vectors instead. Linear SVC achieves very similar, albeit lower, performance values than fastText. This is quite interesting, since both work on an identical form of processed text, with Linear SVC not having the benefit of a custom language model. On the other hand, ML-KNN showed the worst performance on our datasets. While it exhibits high precision, it retrieves fewer labels overall. However, the same or higher precision can generally be achieved with BERT and fastText by raising the confidence threshold while maintaining a higher recall (e.g., increasing the threshold from .2 to .5 for fastText on *BrueLeg*: P 59.2/R 27.0/F_1 37.1).

To further explore the results for the top performing classifier BERT, we examine the distribution of F_1 scores per label. Figure 3(a) shows a significant share of labels in all datasets that cannot be sufficiently learned (i.e., relatively low F_1 values). In addition to *Bank*, this applies to about two-thirds of *BrueLeg*'s labels and the majority of *ZuL*'s labels, although sufficient training examples should be available for most of them. We also compared the performance ranges with the median number of text samples per label (support) in the group (see Figure 3(b)). Except for rare labels, we find no correlation between a higher number of samples and classification success. This suggests that the difficulty of our task goes beyond just finding more training data and is probably related to the consistency of the individual coders and between them.

The structure of the codebook and the definition of the labels seem to be an important factor. *Bank* contains both the fewest samples and the most labels, reflecting a low performance across all classifiers. However, when only top-level labels are considered, the results improve significantly. The same applies to *ZuL* with a larger sample base. While half of the *Bank Base* labels are still problematic, others perform exceptionally well. The significant improvement for the *Base* labels indicates that the top level is split into better delimited topics, while the second level labels may be ambiguous with many commonalities or have low cardinality. Clearly defined groups of codes thus suggest a potential benefit for clustered or multi-stage classification.

These results indicate that modern classification methods can indeed significantly improve the performance of in-dataset code prediction in most cases. However, the overall performance remains too weak to fully automate the coding process. Regardless, ML may still assist coding



Share of labels in F_1 score ranges

Median number of text samples (support) per label in F_1 score ranges

Figure 3. BERT label performance distribution for each dataset.

with code recommendations for predefined segments. Moreover, combining the outputs of multiple classifiers would raise recall and provide ranked suggestions based on the number of votes. For example, combining our three best performing classifiers (BERT, fastText, and Linear SVC) on a majority vote ensemble achieves, for example, a 71.1% recall on *BrueLeg*. We explore this approach further in the next section.

Agree to Disagree: Classifiers Agreement

One way to improve classification results is the combination of multiple classifiers in a majority voting fashion. An important measure of the success of such an approach is the degree of agreement between the classifiers. Standard measures to distill this information into an inter-rater reliability coefficient are Cohen's kappa or Krippendorff's alpha. While both account for the randomness of agreement, interpreting a single number can prove difficult and does not lead to the exact causes behind it.

For further insight into their performance, the overlap of predictions for each label in *BrueLeg* is shown in Figure 4. We have included the three best performing classifiers: BERT, fastText with pre-trained embeddings, and Linear SVC. Each row represents one label of the dataset, with the corresponding samples divided into true-positives and true-negatives. These subsets are subsequently divided into the number of classifiers that would agree with the premise given by the dataset (i.e., ground truth – GT).

While the majority of the negative instances are correctly detected by all three classifiers, we see that 28.9% of the positive samples would not be captured by any of the models. This is influenced by the trade-off made with the chosen target metric, which lowers recall to achieve fewer false positives. Interestingly, the labels with a higher share of disagreement on the GT negative side correlate with an increased agreement on the GT positive side. Upon closer inspection, these also represent the most prominent codes used in the dataset. This suggests that an

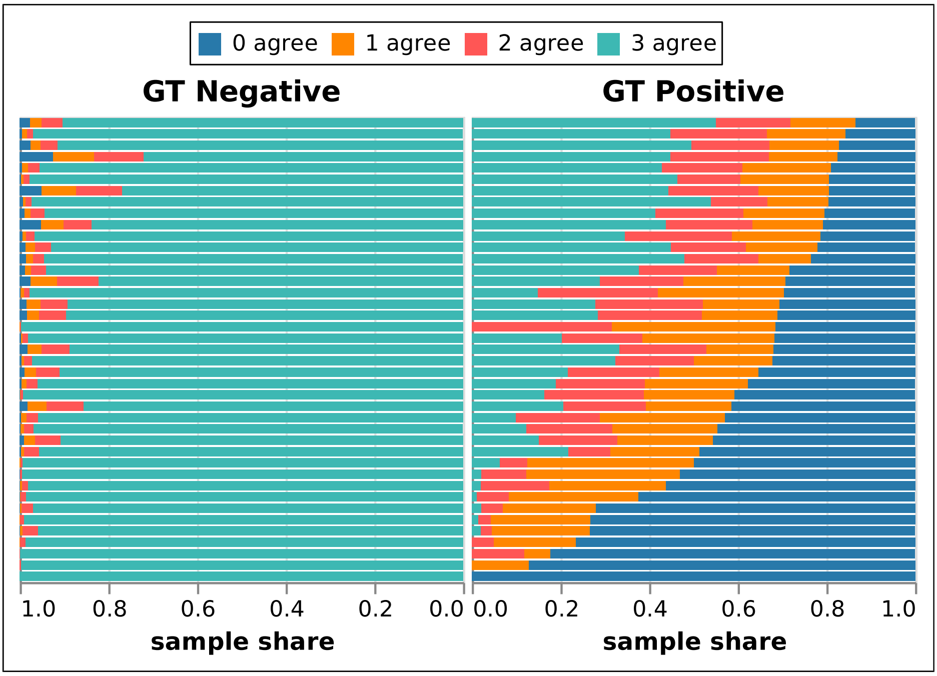


Figure 4. Agreement between multiple classifiers and ground truth (GT) for labels in BrueLeg. For each label (y-axis) it shows the percentages – from total disagreement with the GT (0 agree) to all three classifiers in agreement with the GT (3 agree).

increased number of instances for a label can obscure its meaning and, in the worst case, affect the performance of the classifier. In contrast, we can also find many labels with smaller sample sizes that are correctly predicted by at least two classifiers without increased false positives. In general, a possible explanation for false positives where different classifiers agree with each other would be that the human coder did not persist with that label for all the samples. For example, they may have enough evidence for a label, reach theoretical saturation, and decide not to code additional sections with it.

Overall, we observe a relatively high level of agreement between models, given the characteristics of the dataset. Furthermore, examining cases where multiple classifiers disagree could point analysts to labels that need additional discussion and help with reinterpretation (Drouhard et al., 2017). Labels can be ambiguous, for example, when they define abstract topics, match broad issues, or have been generated by humans with different experiences and viewpoints. Automatically distinguishing among these cases and providing recommendations could be an interesting line of future research.

Codebook Relationship Analysis

In this section, we explore the relationships between individual codes from the BrueLeg and ZuL studies. These two datasets were created and coded by different research groups. Given the thematically similar content but different questions that guided the creation and coding of these datasets, they serve as an excellent proof of concept to demonstrate the effectiveness of our method. Using the trained classifiers, we aim to elicit semantic relationships and, for example, split general codes into subtopics for further analysis.

After collecting label relation confidence scores (lrcs) (see Figure 2 and equation (1)) and empirically determining a threshold to filter out weak relations, a total of 25 codes from BrueLeg show similarities to 30 codes from ZuL. The strongest relationship is found to be 33.4%, and we exclude pairings below 1%. A subset of the resulting relations is depicted in Figure 5.

We find that, for example, the code *"Profession & Career"* from BrueLeg links to ZuL codes such as *"Biography," "Company Career,"* and *"Person/Orientations."* Similarly, the code tagging *"Family & Partnership"* is strongly linked to topics that frame the personal life situation, along with the conflicts and impact of work-life balance on it.

While these connections might also be made by interpreting the code names, other cases show the thematic facets of commonly used and broader codes. One of the most frequent codes in BrueLeg deals with narratives about work and the company (*"Talk about Work & Company"*). Our analysis identifies correlations between this and ZuL codes such as *"Work Content," "Self-Organization," "Employee Structure," "Work Conditions,"* and *"Supervisors."* Since the research underlying ZuL is primarily concerned with time and performance pressures in the work environment, specific granular codes were used to capture each aspect of this topic. This allows further insights when looking at the corresponding codes of BrueLeg such as "Performance, Stress, Work Environment." The code can thus be broken down into sources of stress such as supervisors or working conditions, the consequences on the work-life balance, and the individual's handling of these situations.

These cross-dataset relations could contribute to the analysis by, for example, applying similar codes from other studies, providing an initial coding, and highlighting relevant sections to bootstrap the coding process. Transferred to large interview documents, this may provide structure to facilitate navigation and retrieval (Zhang et al., 2019). It also allows researchers to apply interpretations from other studies and gain new insights into their data. Our initial findings highlight the value and potential of expanding investigation methods and reusing this kind of research data for secondary analysis or qualitative meta-analysis.

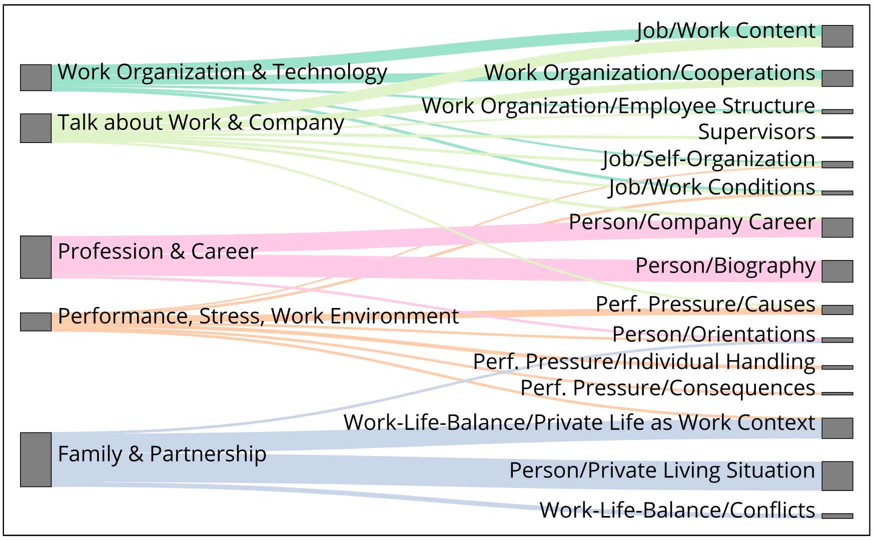

Work Organization & Technology
Talk about Work & Company
Profession & Career
Performance, Stress, Work Environment
Family & Partnership
Job/Work Content
Work Organization/Cooperations
Work Organization/Employee Structure
Supervisors
Job/Self-Organization
Job/Work Conditions
Person/Company Career
Person/Biography
Perf. Pressure/Causes
Person/Orientations
Perf. Pressure/Individual Handling
Perf. Pressure/Consequences
Work-Life-Balance/Private Life as Work Context
Person/Private Living Situation
Work-Life-Balance/Conflicts

Figure 5. Extracted code relations between BrueLeg (left) and ZuL (right) datasets.

Discussion

In this work, we analyzed the applicability of supervised multi-label classification models to learning from coded qualitative research data and relating the codebooks of different studies to each other. Our results on three real-world social science datasets highlight some difficulties arising from the subjectivity of the task. The coherence of labels required for automated classification depends on the research task and the definition of the codebook. With multiple annotators working on larger datasets, the quality and consistency of coding is influenced by their experience and different interpretations. Moreover, broader label scopes can introduce bias into the classifier, and the amount of text selected to provide enough context for specific labels increases the chance of capturing additional topics. The BERT model proved to be more resilient to these challenges and showed superior performance to linear models. Its extensive language model and contextual awareness (e.g., capturing synonyms) could improve our task performance. Although fully automatic coding in the process of qualitative analysis is likely not desired, nor is the performance of the models so far sufficient for this purpose, these classifiers could be used for coding recommendations. In addition, they offer the potential to detect possible ambiguities in codes or different interpretations among coders.

We proposed initial steps to link and retrieve relevant content from multiple datasets by exploiting the trained classifiers. Experimental results suggest that the approach helps to identify commonalities between the codebooks of studies that were coded independently and with different research questions. While this metadata is typically only created and used during the study, our findings highlight its reuse value for exploring further subtopics in recorded data. Identifying commonalities between the codebooks of different studies could allow benefiting from previous experience, reuse of existing research data, and bootstrapping the coding process. A more accurate extraction of semantic knowledge embedded in archived metadata could help create new insights and facilitate retrieval and secondary content analysis.

Our methodology could facilitate and enhance data integration. Using machine learning techniques, analysts should be able to query qualitative data libraries (e.g., (University of Essex, 2023)) to retrieve additional relevant content based on the semantic inference of their own codebook, enabling richer and more comprehensive analyses (Hienert et al., 2019). Our analysis and grounded theory are mostly geared toward well-resourced approaches like Big Qual (Brower et al., 2019), which usually involve a team of researchers and are in a better position to generate their own initial representative dataset and codebook from which the ML can then produce the linking. Nevertheless, numerous qualitative researchers operate independently, the majority of whom are engaged in small data research (Kitchin & McArdle, 2016). In such cases, limited resources might also translate into the need to use third-party datasets over which there is restricted control, for example, in terms of scope, temporality, and size. Here, the proposed methodology can help in transferring knowledge from richer domains and bigger studies. Moreover, these smaller studies can be assembled with the help of machine learning techniques such as the one proposed in this work. These constructed collections offer prospects for exploring fresh research inquiries by leveraging distinctions among the studies. Details concerning the "context" of each study, such as its particular focus, unit of investigation, sample characteristics, geographic and temporal setting, researchers' backgrounds in social sciences, and their data collection or analysis methodologies, serve as metadata that can shape a comparative framework (Davidson et al., 2019).

Given the limitations of simple metrics in accurately assessing potential label relationships, a manual evaluation of classification results becomes necessary to identify uncertainties in the ground truth and to re-evaluate false positives. Furthermore, our datasets lack information regarding the assignment of documents to the individuals coding them. Developing separate models for each coder would allow better identification of disparities in code interpretation and facilitate the early detection of ambiguities during the coding process. In the future, we aim to expand our research by incorporating prior knowledge from other studies into our classifiers and further experimenting with knowledge transfer. We will also explore promising approaches for exploiting meta-information from previous QDAs. For example, we could combine semantically similar codes and amalgamate the collective knowledge extracted from various studies into a comprehensive classification model. When applied to a new collection of documents, this approach would provide an initial categorization and generate a content index within each document. In general, our research aims to improve the efficiency of the qualitative data analysis process by helping analysts benefit from previous interpretation efforts and gain new insights into their data.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This paper is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101021866 (CRiTERIA).

ORCID iD

Sergej Wildemann https://orcid.org/0000-0001-9216-4253

Notes

1. https://www.maxqda.com/products
2. https://atlasti.com/atlas-ti-desktop
3. https://lumivero.com/products/nvivo/

References

Bakharia, A., Bruza, P., Watters, J., Narayan, B., & Sitbon, L. (2016). Interactive topic modeling for aiding qualitative content analysis. In Proceedings of the 2016 ACM Conference on Human Information Interaction and Retrieval, CHIIR'16, 13-17 March, 2016, Carrboro, NC, USA (Vol. 2016, pp. 213–222). ACM.

Baumer, E. P. S., Mimno, D. M., Guha, S., Quan, E., & Gay, G. K. (2017). Comparing grounded theory and topic modeling: Extreme divergence or unlikely convergence? *Journal of the Association for Information Science and Technology*, *68*(6), 1397–1410.

Bernard, H. R. (2013). *Social research methods: Qualitative and quantitative approaches*. Sage.

Brower, R. L., Jones, T. B., Osborne-Lampkin, L., Hu, S., & Park-Gaghan, T. J. (2019). Big qual: Defining and debating qualitative inquiry for large data sets. *International Journal of Qualitative Methods*, *18*. DOI:10.1177/1609406919880692

Chan, B., Schweter, S., & Möller, T. (2020). German's next language model. *CoRR*. DOI:10.48550/arXiv.2010.10906

Charmaz, K. (2006). *Constructing grounded theory: A practical guide through qualitative analysis*. Sage.

Chen, N., Drouhard, M., Kocielnik, R., Suh, J., & Aragon, C. R. (2018). Using machine learning to support qualitative coding in social science: Shifting the focus to ambiguity. *ACM Transactions on Intelligent Systems and Technology*, *8*(2), 1–20, 9. DOI:10.1145/3185515

Crowston, K., Allen, E. E., & Heckman, R. (2012). Using natural language processing technology for qualitative data analysis. *International Journal of Social Research Methodology*, *15*(6), 523–543. DOI:10.1080/13645579.2011.625764

Crowston, K., Liu, X., & Allen, E. (2010). *Machine learning and rule-based automated coding of qualitative data. In Navigating Streams in an Information Ecosystem* - Proceedings of the 73rd ASIS&T Annual Meeting, ASIST 2010, Volume 47 of Proceedings of the Association for Information Science and Technology, 22-27 Oktober, 2010, Pittsburgh, PA, USA (pp. 1–2). Wiley.

Cui, W., Wang, Y., Zou, B., & Zou, Z. (2019). Soil heavy metal qualitative classification model based on hyperspectral measurements and transfer learning. *Spectroscopy and Spectral Analysis*, *39*(8), 2602–2607. DOI:10.3964/j.issn.1000-0593(2019)08-2602-06

Davidson, E., Edwards, R., Jamieson, L., & Weller, S. (2019). Big data, qualitative style: A breadth-and-depth method for working with large amounts of secondary qualitative data. *Quality and Quantity*, *53*, 363–376. DOI:10.1007/s11135-018-0757-y

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, *NAACL-HLT* 2019, June 2-7, 2019, Minneapolis, Minnesota, USA, Volume *1* (Long and Short Papers) (pp. 4171-4186). Association for Computational Linguistics.

Drouhard, M., Chen, N., Suh, J., Kocielnik, R., Araya, V. P., Cen, K., Zheng, X., & Aragon, C. R. (2017). Aeonium: Visual analytics to support collaborative qualitative coding. In IEEE Pacific Visualization Symposium, PacificVis 2017, 18-21 April, 2017, Seoul, Korea (South) (pp. 220–229). IEEE Computer Society.

Dunkel, W., & Kratzer, N. (2016). *Zeit- und Leistungsdruck bei Wissens- und Interaktionsarbeit: Neue Steuerungsformen und subjektive Praxis* (1st ed.). Nomos Verlagsgesellschaft mbH & Co. KG.

Ebesu, T., & Fang, Y. (2017). *Neural citation network for context-aware citation recommendation*. In Proceedings of the 40th International ACM SIGIR Conference on Research And Development in Information Retrieval, SIGIR '17, 7-11 August, 2017, Shinjuku, Tokyo, Japan (pp. 1093–1096). Association for Computing Machinery.

Flick, U. (2013). *The Sage handbook of qualitative data analysis*. Sage.

Freitas, F., Ribeiro, J., Brandão, C., de Souza, F. N., Costa, A. P., & Reis, L. P. (2017). In case of doubt see the manual: A comparative analysis of (self) learning packages qualitative research software. In International Symposium on Qualitative Research, ISQR 2017, July 12-14, 2017, Salamanca, Spain (pp. 176–192).

Glaser, B. G., Strauss, A. L., & Strutzel, E. (1968). The discovery of grounded theory; strategies for qualitative research. *Nursing research*, *17*(4), 364. DOI:10.2307/2094063

He, Z., & Schonlau, M. (2020). Automatic coding of text answers to open-ended questions: Should you double code the training data? *Social Science Computer Review*, *38*(6), 754–765. DOI:10.1177/0894439319846622

Hienert, D., Kern, D., Boland, K., Zapilko, B., & Mutschke, P. (2019). *A digital library for research data and related information in the social sciences*. In Proceedings of the 18th Joint Conference on Digital Libraries, JCDL '19, 2-6 June 2019, Champaign, IL, USA (pp. 148–157). IEEE Press.

Jiang, J. A., Wade, K., Fiesler, C., & Brubaker, J. R. (2021). Supporting serendipity: Opportunities and challenges for human-ai collaboration in qualitative analysis. *Proceedings of the ACM on Human-Computer Interaction*, *5*(CSCW1), 1–23. DOI:10.1145/3449168

Joulin, A., Grave, É., Bojanowski, P., & Mikolov, T. (2017). *Bag of tricks for efficient text classification*. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, April 3-7, 2017, Valencia, Spain *(Vol. 2,* pp. 427–431).

Kaufmann, A., Barcomb, A., & Riehle, D. (2020). *Supporting interview analysis with autocoding*. In Proceedings of the 53rd Hawaii International Conference on System Sciences, HICSS 2020, Maui, Hawaii, USA, January *7-10*, 2020 (pp. 752–761). ScholarSpace. DOI:10.24251/hicss.2020.094

Kitchin, R., & McArdle, G. (2016). What makes big data, big data? Exploring the ontological characteristics of 26 datasets. *Big Data & Society*, *3*(1). DOI:10.1177/2053951716631130

Knoblauch, H. (2013). Qualitative methods at the crossroads: Recent developments in interpretive social research. *Forum Qualitative Sozialforschung/Forum: Qualitative Social Research*, *14*(3), 257–270, 12. DOI:10.17169/fqs-14.3.2063

Kratzer, N., Menz, W., & Pangert, B. (2015). *Work-Life-Balance - eine Frage der Leistungspolitik. Analysen und Gestaltungsansätze* (1st ed.). Wiesbaden: Springer VS.

Kratzer, N., Menz, W., Tullius, K., & Wolf, H. (2019). *Legitimationsprobleme in der Erwerbsarbeit: Gerechtigkeitsansprüche und Handlungsorientierungen in Arbeit und Betrieb*. Nomos Verlag.

Liew, J. S. Y., McCracken, N., Zhou, S., & Crowston, K. (2014). Optimizing features in active machine learning for complex qualitative content analysis. In *Proceedings of the Workshop on Language Technologies and Computational Social Science@ACL* 2014, June *26*, 2014, Baltimore, Maryland, USA (pp. 44–48). Association for Computational Linguistics.

Marathe, M., & Toyama, K. (2018). *Semi-automated coding for qualitative research: A user-centered inquiry and initial prototypes*. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI 2018, 21-26 April, 2018, Montreal, QC, Canada (p. 348). ACM.

Paredes, P., Ferreira, A. R., Schillaci, C., Yoo, G. R., Karashchuk, P., Xing, D., Cheshire, C., & Canny, J. F. (2017). Inquire: Large-scale early insight discovery for qualitative research. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing, CSCW 2017,* February 25 - March 01, 2017, Portland, OR, USA (pp. 1562–1575). ACM. DOI:10.1145/2998181.2998363

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1 - 6 June, 2018, New Orleans, Louisiana, USA (Vol. 1, pp. 2227–2237). Association for Computational Linguistics.

Pruksachatkun, Y., Phang, J., Liu, H., Htut, P. M., Zhang, X., Pang, R. Y., Vania, C., Kann, K., & Bowman, S. R. (2020). Intermediate-task transfer learning with pretrained language models: When and why does it work? In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 5 - 10 July, 2020, Online (pp. 5231–5247). Association for Computational Linguistics.

Rietz, T., & Maedche, A. (2021). Cody: An ai-based system to semi-automate coding for qualitative research. In Y. Kitamura, A. Quigley, K. Isbister, T. Igarashi, P. Bjørn, & S. M. Drucker (Eds.), *CHI21: CHI Conference on Human Factors in Computing Systems, Virtual Event/Yokohama*, Japan, 8-13 May, 2021 (Article 394, pp. 1 - 14). ACM.

Strauss, A., & Corbin, J. M. (1997). *Grounded theory in practice*. Sage.

Taroni, J. N., Grayson, P. C., Hu, Q., Eddy, S., Kretzler, M., Merkel, P. A., & Greene, C. S. (2019). Multiplier: A transfer learning framework for transcriptomics reveals systemic features of rare disease. *Cell systems*, *8*(5), 380–394. DOI:10.1016/j.cels.2019.04.003

Tenney, I., Das, D., & Pavlick, E. (2019). *BERT rediscovers the classical NLP pipeline*. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, July 28 - August 2, 2019, Florence, Italy (pp. 4593–4601). Association for Computational Linguistics. DOI:10.18653/v1/P19-1452

Tierney, P. J. (2012). A qualitative analysis framework using natural language processing and graph theory. *International Review of Research in Open and Distributed Learning*, *13*(5), 173–189. DOI:10.19173/irrodl.v13i5.1240

Tsoumakas, G., & Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, *3*(3), 1–13. DOI:10.4018/978-1-60566-058-5.ch021

University of Essex (2023). *UK data archive*. https://www.data-archive.ac.uk/. Accessed September 08, 2023.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. (2019). Superglue: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 32). Curran Associates, Inc.

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, *3*(1), 1–40, 9. DOI:10.1186/s40537-016-0043-6

Wildemann, S., Niederée, C., & Elejalde, E. (2023). Migration reframed? A multilingual analysis on the stance shift in europe during the Ukrainian crisis. *Proceedings of the ACM Web Conference*, *2023*, 2754–2764. DOI:10.1145/3543507.3583442

Xie, Y., Sun, Y., & Bertino, E. (2021). Learning domain semantics and cross-domain correlations for paper recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'21, July 11-15, 2021, Virtual Event, Canada* (pp. 706–715). Association for Computing Machinery.

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing [Review Article]. *IEEE Computational Intelligence Magazine*, *13*(3), 55–75. DOI: 10.1109/MCI.2018.2840738

Zhang, M.-L., & Zhou, Z.-H. (2007). ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, *40*(7), 2038–2048. DOI:10.1016/j.patcog.2006.12.019

Zhang, R., Guo, J., Fan, Y., Lan, Y., & Cheng, X. (2019). Outline generation: Understanding the inherent content structure of documents. In B. Piwowarski, M. Chevalier, É. Gaussier, Y. Maarek, J. Nie, & F. Scholer (Eds.), Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21 - 25, 2019 (pp. 745–754). ACM. DOI:10.1145/3331184.3331208

Zhao, C., Li, C., Xiao, R., Deng, H., & Sun, A. (2020). Catn: Cross-domain recommendation for cold-start users via aspect transfer network. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, July 25-30, 2020, Virtual Event, China (pp. 229–238). ACM. DOI:10.1145/3397271.3401169

Zhu, C., Cheng, Y., Gan, Z., Sun, S., Goldstein, T., & Liu, J. (2020). Freelb: Enhanced adversarial training for natural language understanding. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April, 2020. OpenReview.net.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, *109*(1), 43–76. DOI:10.1109/JPROC.2020.3004555

Author Biographies

Sergej Wildemann is a research assistant and Ph.D. candidate at L3S Research Center, Hannover, Germany. He completed his master's degree in Computer Science at Leibniz University Hannover in 2018. His research interest expands to digital libraries, temporal information retrieval, and social media analysis.

Claudia Niederée received the master's degree in computer science and the Ph.D. degree in computer science from the Technische Universität Hamburg, Hamburg, Germany, in 1995 and 2002, respectively. She is currently a Research Group Leader with the L3S Research Center, Hannover. She has published more than 100 scientific articles and papers. She is involved in the European project CRiTERIA, where she and her team are working on advanced analysis technologies and tools that are tailored to new comprehensive threat indicators developed as part of the CRiTERIA methodology. Her main research interests are diversity in knowledge, digital forgetting, social media analysis, entity-centric technology, and AI-based systems.

Erick Elejalde received the M.Sc. and Ph.D. degrees in computer science from the University of Concepción, Concepción, Chile, in 2013 and 2018, respectively. He is currently a Researcher with the L3S Research Center, Leibniz University Hannover, Hannover, Germany. His research interests include computational social science, user modeling, and social networks analysis.