Mapping Work Task Descriptions from German Job Ads on the O*NET Work Activities Ontology

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

Abstract

This work addresses the challenge of extracting job tasks from German job postings and mapping them to the fine-grained work activities classification in the O*NET labor market ontology. By utilizing ontological data with a Multiple Negatives Ranking loss and integrating a modest volume of labeled job advertisement data into the training process, our top configuration achieved a notable precision of 70% for the best mapping on the test set, representing a substantial improvement compared to the 33% baseline delivered by a general-domain SBERT. In our experiments the following factors proved to be most effective for improving SBERT models: First, the incorporation of subspan markup, both during training and inference, supports accurate classification, by streamlining varied job ad task formats with structured, uniform ontological work activities. Second, the inclusion of additional occupational information from O*NET into training supported learning by contextualizing hierarchical ontological relationships. Third, the most significant performance improvement was achieved by updating SBERT models with labeled job ad data specifically addressing challenging cases encountered during pre-fine-tuning, effectively bridging the semantic gap between O*NET and job ad data.

Keywords: Text Mining, Work Task Classification, Domain Adaptation

1. Introduction

What people do at work, and the evolution of work activities between and within professions over time are of great interest to social scientists and labor market stakeholders. Job advertisements serve as an excellent data source for addressing these questions as beyond specifying a job's profession or industry, they provide detailed descriptions of tasks and responsibilities associated with specific positions at a given time. Our research aims to facilitate investigations into these questions by normalizing free text formulations to ontological categories.

Our first objective is to extract information regarding individual job tasks from job advertisements. Through standardization of the extracted job tasks and their connectivity to other data sources, we seek to enable statistical analyses and comparative research. Our second objective is thus to provide a method for mapping extracted job tasks to established labor market ontologies or taxonomies.

This paper focuses on German-language job ads from the Swiss Job Market Monitor (Buchmann et al., 2023), a longitudinal and representative research dataset from Switzerland. We select O*NET[1] as mapping ontology for its unique hierarchical classification of work activities, ranging from fine-grained (about 2,000 classes) to more aggregated categories (37). O*NET also offers insights into task-occupation associations, along with a mapping of tasks to work activities, making it the primary ontology of its kind.

To achieve our first goal, the extraction of job tasks from job postings, we must deal with the diverse ways in which job tasks are expressed. We develop a framework for the recognition of job tasks, job titles, and the relevant components of tasks, such as work activities, objects, scope of responsibility, and contextual information. We aim to implement task recognizers that extract task text spans and standardize the various ways job tasks and duties are presented in job ads by further segmenting and classifying their components.

To reach our second goal, ontology mapping, we train sentence-level semantic vector representation models, facilitating the semantic retrieval of O*NET work activities corresponding to job ad tasks. To this end, we harness ontological data with a limited amount of labeled job ad data, ensuring optimal vector representations for semantic search. Furthermore, we employ subspan markup within the training and inference processes to incorporate structural information related to work tasks.

Our contributions in this paper encompass the definition of job tasks and their internal components, the training of domain-specific language representation models at the sentence level for semantic similarity retrieval, and the creation of a gold standard dataset for evaluating task extraction from German-language job ads and their mapping to O*NET.[2]

In the following, we discuss related work in Section 2 and describe our experimental data and

[1] https://www.onetonline.org/

[2] Data created for this paper is available via DOI 10.5281/zenodo.10868835

LREC-COLING 2024, pages 11049–11059
20-25 May, 2024.

pipeline in Section 3. The approaches and experiments for task span extraction are explained in Section 4, and our techniques and experiments for mapping job tasks to O*NET are presented in Section 5. Our key findings are summarized in Section 6.

2. Related Work

Information extraction from job ads, with a primary focus on occupations and skill requirements, has long been explored in social science, however often on proprietary data with not fully documented methods (Atalay et al., 2020; Deming and Kahn, 2018; Acemoglu et al., 2022). More recent approaches involve distant supervision using ontologies like ESCO (Zhang et al., 2022; Decorte et al., 2022). Notably, Zhang et al. (2023) employ ESCO to pretrain a multilingual language model. They introduce a relation prediction objective, alongside the standard Masked Language Modeling objective, to distinguish between hierarchically related, otherwise related, and unrelated ontological concepts. This approach yields state-of-the-art results in various sequence labeling and classification problems.

Decorte et al. (2023) aim to detect skills and associate them with an ontology through fine-grained multi-label classification. They create a synthetic training dataset by obtaining example sentences from job ads for specific ontology concepts querying large language models. They then learn semantic vector representations by combining these examples with concept labels, employing a Multiple Negatives Ranking (MNR) loss.

In our previous work, we approach fine-grained skill classification similarly to Decorte et al. (2023), leveraging ontology data with MNR, but employ fine-grained extractors for skill components and utilize such structural data alongside contextual sentence representations, in contrast to Decorte's focus on entire sentences (Gnehm et al., 2022).

3. Experimental Data and Pipeline

SJMM, the Swiss Job Market Monitor[3], is a multilingual, longitudinal, and representative research dataset of job ads from Switzerland. In our experiments, we focus on German-speaking job ads from 1990 onwards (n=480k ads).

O*NET, the Occupational Information Network, is an online database that offers access to an extensive collection of information, including tasks, work activities, or skills related to more than 900 distinct occupations spanning the entire U.S. economy. For our research, of primary interest is the available classification of over 18,000 Work Tasks

[3]Available under https://www.swissubase.ch

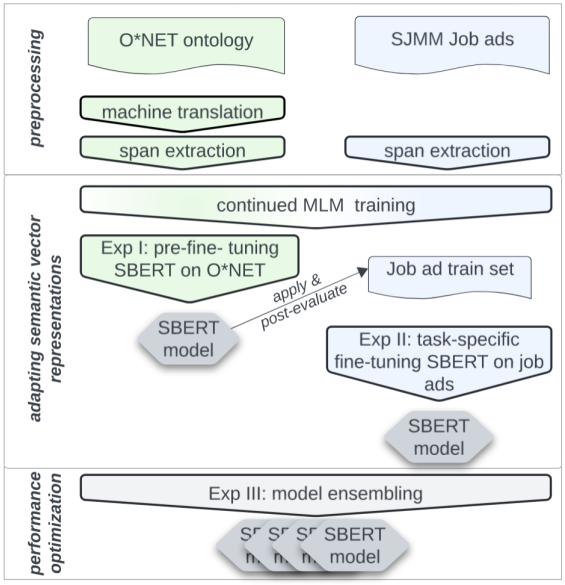

Figure 1: Job ad task mapping pipeline. All SBERT models are evaluated on job ad dev or test data (not shown here).

into a hierarchical structure of Work Activities. This taxonomy comprises 37 General Work Activities (GWA), 332 Intermediate Work Activities (IWA), and over 2,000 Detailed Work Activities (DWA). Most tasks (78.5%) are assigned to a single DWA, but 18.3% are connected to two DWAs and 3.2% to three to five DWAs. For instance, the Task "Assign schedules to work crews" is assigned to the DWA "Plan employee work schedules", as well as to the DWA "Assign duties or work schedules to employees". Most GWAs encompass up to 500 tasks, but there is one GWA ("Handling and Moving Objects"), comprising around 3,000 tasks. Also relevant to our research is the connection between the 900 occupation classes and their typical tasks.

Pipeline: We undertake the following key steps to enable a mapping from the SJMM job ad tasks to the O*NET work activities (see Figure 1): As a ***preprocessing***, O*NET data is *machine-translated* from English to German. Then, both datasets undergo a *two-step span extraction*, to segment single work task spans, and to identify within them specific subspans, such as work objects or responsibility roles (see Section 4). Next, we undertake several steps to ***adapt semantic vector representations*** for the similarity-based retrieval of O*NET activities for job ad task spans (outlined in Section 5). We begin with *continued masked language modeling (MLM)* of transformer-based language models, utilizing both job ad and ontology task spans. Subsequently, we *pre-fine-tune sentence-level representations (SBERT)* on ontology data, mimicking the task of retrieving O*NET work activities for job ad tasks by learning similarities between O*NET

work activities and ontological tasks (Section 5.1.1). We then apply and evaluate these initial models for retrieving O*NET work activities for job ad tasks. From this post-evaluated data, we identify challenging cases to create task-specific training data, that we utilize for *task-specific model fine-tuning* (see Section 5.1.2). Finally, we ***optimize performance*** by employing model ensembling.

For the first step of translating monolingual English O*NET data into German, we utilized DeepL[4], a state-of-the-art commercial translation system. We verified and, where necessary, corrected translations for all 37 GWA labels and 332 IWA labels, as well as shorter text spans, since translation errors often arise from a lack of context. For instance, "Train personnel" was mistranslated as "Persönlicher Zug" (German for a personal train or personal trait), which we corrected to the more appropriate "Personal schulen". Random inspections suggest that task descriptions, on average 100 characters long, were translated correctly. Consequently, we have confidence in the quality of our translations and anticipate minimal impact from any remaining errors. The experimental settings and results for the subsequent pipeline steps are outlined in the following chapters.

4. Task Span Extraction

We apply a two-step process for the extraction and analysis of task spans in job ads. The first step identifies text segments that specify a single task. This is not trivial because task descriptions are presented in many variants, including lengthy and elaborate formulations, short dense bullet lists, and generic or job-specific descriptions. Because occupation names are a helpful contextualization of work activities, our task segmentation also recognizes job title mentions in job postings.

Once a task span describing a single work activity has been identified, it undergoes a second segmentation step where specific subspans are identified and categorized. Namely: job title ([BJT],[EJT])[5]; work activity, work object, work object/activity as single words ([BOA],[EOA]); responsibility role ([BRR],[ERR]); work context ([BCO],[ECO]).

Examples from O*NET with in-text markup: "[BOA] Collect evidence [EOA] [BCO] for court proceedings [ECO]", "[BRR] Participate [ERR] in [BOA] personnel decisions [EOA]". Examples from translated ads: "[BRR] Ensuring [ERR] [BOA] the telephone accessibility of the office [EOA] [BCO] of the teaching association [ECO]", "[BRR] Contribute [ERR] to [BOA] in-house continuing education [EOA] [BCO] in the area of anesthesia [ECO]".

	Gold Task Spans			Model Task Spans		
	P	R	F1	P	R	F1
Job Title	88.9	91.7	90.3	93.1	89.8	91.4
Activity	92.8	92.8	92.8	94.7	93.1	93.9
Object	86.4	85.9	86.2	85.1	89.8	87.4
Object/Act.	85.5	87.5	86.5	87.6	80.7	84.0
Resp. Role	91.9	91.1	91.5	87.6	89.3	88.5
Context	69.9	80.3	74.8	75.5	77.6	76.6
AVERAGE	85.9	88.2	87.0	87.3	86.7	86.9

Table 1: Precision (P), Recall (R), and F1-Score (F1) of subspan categories on the test set with gold task spans and model-predicted task spans.

4.1. Annotation, Models, and Results

For both extraction steps, we bootstrapped our first model on a small, manually annotated sample (64 ads for task span extraction, 300 task spans for subspan extraction) by training a transformer-based NER-style tagging model with spaCy[6]. For the embeddings, we used jobGBERT[7], a German transformer model adapted to job ads by further Masked Language Modeling pre-training of GBERT (Chan et al., 2020). Then, we iteratively expanded our training set in several annotation rounds, by correcting model predictions on new data. To this end, we used the annotation tool *prodigy*.[8] For the job ad sampling, we made sure that all broader occupation classes were represented. This process yielded a dataset of 1,675 ads for task span extraction and 7,234 task spans for subspan extraction. Both datasets were divided into train, dev, and test sets using an 80-10-10 split. On the test set, our task extraction evaluated with strict boundary matches on span level achieved 82.5% F1-Score (83.1% Precision, 81.8% Recall). We observed that several errors were due to boundary mismatches, and hence, we also evaluated the recognition quality on the token level, which resulted in 92.3% F1-Score (93.8% Precision, 90.8% Recall).

For subspan extraction, we created an additional test set (567 spans) with span boundaries segmented by the span extractor model (instead of human-annotated gold data) to evaluate its capability to deal with imperfectly segmented spans. The results in Table 1 show that the subspan model is robust to slightly imperfect segmentations. The overall performance of the subspan recognizer has 87% F1-Score, but the most relevant and frequent subspan category Activity reaches 93-94% F1-Score.

[4] https://www.deepl.com

[5] Tags in brackets such as [BJT] (begin of job title) or [EJT] (end of job title) are used when serializing the structural markup into running text.

[6] https://spacy.io We used the default settings of the components *spacy-transformers.TransformerModel.v1* and *spacy.TransitionBasedParser.v2*

[7] Available on Hugging Face: jobGBERT

[8] https://prodi.gy/

5. Mapping to O*NET ontology

The goal of this work is to enable a comprehensive depiction of the duties outlined in a job posting. Particularly in the context of social science research, the opportunity to examine the combination of different activities within jobs, and to construct and explore work activity profiles, is intriguing. Consequently, we aim to map every single extracted job task span to an ontological work activity.

We focus on the Intermediate Work Activities (IWA) level of O*NET, comprising 330 distinct categories. This level provides detailed insights while maintaining independence from occupation-specific categorizations. This allows, firstly, for easier comparison of task profiles across diverse occupations. Secondly, disparities likely exist between the US and Swiss labor markets regarding the prevalence of occupations and associated tasks. However, the high level of abstraction of IWAs (most IWAs are linked to many different occupations in O*NET) suggests compatibility across different labor markets.

Our mapping approach resembles an information retrieval scenario, where we aim to identify the most relevant O*NET work activities for a given job ad task query. To tackle this challenge, we employ semantic similarity matching, relying on cosine similarities between the job ad tasks and the textual descriptions of the candidate work activity nodes. While we extract and match individual job ad tasks, it is worth noting that they may inherently correspond to multiple O*NET activities. This raises broader questions about the alignment of ontology with real-world data and the effectiveness of bridging the gap between ontology and empirical data. In summary, this mapping can also be viewed as an unsupervised, fine-grained multi-label classification problem.

5.1. Adapting Semantic Vector Spaces

To establish a shared vector space for the two types of texts, job postings and O*NET descriptions, where corresponding tasks are closely aligned, we are experimenting with various approaches to pre-training and fine-tuning existing large language models.

MLM: Further pretraining with Masked Language Modeling (MLM), as introduced by Devlin et al. (2019), has proven effective in adapting language models to specialized domains (Gururangan et al., 2020). We build on the existing transformer-based model jobGBERT[9] that has been pre-trained on German-speaking job ad texts. We continue MLM pretraining using the extracted task spans and job titles from job ads as well as the task descriptions, work activities, and occupation class titles from the translated O*NET. We utilize 1.7 million unique spans extracted from job ads and 22,000 spans from O*NET. We oversample ontology spans by factor 5 and resample job ad spans to ensure equal representation across all years from 1990 to 2022. In total, our training data comprises 2 million spans. Detailed training parameters are provided in the appendix. This step adapts the language model specifically to our in-domain text snippets that express work activities.

MNR: Reimers and Gurevych (2019) have shown that MLM-trained BERT embeddings are not suitable for semantic similarity comparisons of sentences and suggested several fine-tuning tasks for improving the vector space of so-called Sentence-BERT models (SBERT). We employ Multiple Negatives Ranking (MNR) loss to train domain-specific SBERT models for improved semantic similarity lookup. For our SBERT models, Siamese networks are trained on sentence pairs: Every positive pair (with a semantically related sentence) is contrasted to several negative pairs (unrelated sentences), and the model learns to assign a higher similarity score to positive pairs. The negative pairs are by default sampled randomly from the same batch, or specifically selected as challenge pairs resembling positive cases (hard negatives, indicated with **HN** in model acronyms).

5.1.1. Pairing O*NET Descriptions

In MNR training, we first exploit the available ontological data to ***pre-fine-tune*** SBERT models. While labeled data for our end task – determining ontological work activities (IWAs) for tasks in job positing – is unavailable, we leverage existing ontological data to simulate a very similar task, assuming it enables highly effective pre-fine-tuning. We utilize hierarchical relationships in O*NET and incorporate occupational information to create the necessary positive pairs of related or similar sentences for MNR training (and use in-batch sampling to generate dissimilar pairs). We experiment with three different settings of how to utilize the O*NET data.

In the **T2Act** (task to activity pairing) setting, we pair O*NET tasks with their respective work activities and lower-level work activity classes with higher-level ones (DWA to IWA, DWA to GWA, and IWA to GWA). This results in 88,680 pairs in total.

Additionally, we combine occupation-associated work task information in two ways:

First, inspired by the language used in job advertisements, we incorporate occupation classes as context for work tasks when combining them with work activities, **T*as*Occ2Act** (task as done in occupation to activity pairing). This serves the purpose of disambiguating tasks. For example, “keeping records as a medical secretary” is positively paired

[9] Available on Hugging Face: jobGBERT

Training Data Setting	Examples
T2Act	**s1**: Monitor permit requirements for updates. **s2**: Monitor external affairs, trends, or events.
T*as*Occ2Act	**s1**: Order drugs or devices necessary for study completion as Clinical Research Coordinator. **s2**: Purchase materials, equipment, or other resources.
T*as*Occ2Act$_{MU}$	**s1**: [BOA] Order drugs or devices [EOA] [BCO] necessary for study completion [ECO] as Clinical Research Coordinator. **s2**: [BOA] Purchase materials, equipment, or other resources. [EOA]
T*or*Occ2Act	**s1**: Industrial Operations Manager **s2**: Monitor external affairs, trends, or events.
ad spans	**s1**: Office work **s2**: Perform administrative or clerical activities.
ad spans, HN	**s1**: Caring for our residents **s2**: Assist individuals with special needs. **HN**: Care for plants or animals.

Table 2: MNR examples for different training data settings. If not specified otherwise (HN), random sampling of negatives is used (not shown here). Examples with markup provided in setting TasOcc2Act$_{MU}$.

with "preparing medical documents", while "keeping records as an accountant" is paired with "preparing financial documents". This approach not only disambiguates but also augments our training data because tasks related to several occupations lead to 332k additional pairs.

Second, we combine occupations with work activities of their respective tasks **T*or*Occ2Act**, (occupation to activity pairing, next to task to activity pairing) bringing occupational classes closer to the activities they perform in the semantic vector space. For instance, "Medical secretary" is paired with "preparing medical documents". Lower-level work activities in the hierarchy are occupation-specific, whereas higher-level activities become progressively more abstract and occupation-independent. Consequently, we associate occupations only with activities up to the IWA level, which leads to a total of 213k pairs.

Since the amount of data differs between the different settings, for comparability we train in all three settings for the same number of steps (138,550), which corresponds to 50 epochs for setting T2Act, roughly 10 epochs for setting T*as*Occ2Act, and 25 epochs for setting T*or*Occc2Act. Detailed training parameters are provided in the appendix.

5.1.2. Pairing Job Ad Spans with O*NET

An additional step in MNR training is then ***task-specific fine-tuning***[10]: We create training data specific to our mapping problem by using pre-fine-tuned models on our target data (job ad task spans) and evaluate the IWA suggestions provided. During this phase, we evaluate roughly 8k pairs by selecting a) very generic or frequent job ad terms; b) representative cluster terms; c) low-similarity terms; and d) terms closely associated with IWAs. By including both generic terms and low-similarity terms, we aim to include especially difficult cases in our data. We use this evaluation data as training data to update pre-fine-tuned models in two ways: We utilize the positively evaluated candidates as positive pairs for training, essentially reinforcing the model. These positive pairs are combined with random negative samples in the same batch (in setting **+ Ads**). Alternatively, we also incorporate negatively evaluated candidates, which, since displaying high similarities, represent hard negatives (in setting **+ AdsHN**). For these fine-tuning experiments, we apply the same training parameters as above but, due to the smaller amount of training data, limit the training to 10 epochs.

5.2. Subspan Markup

We explore the influence of subspan markup, as detailed in Section 4. Specifically, we mark the start and end of subspans for Responsibility Roles ([BRR], [ERR]), Context ([BCO], [ECO]), and a combination of work activities and objects ([BOA], [EOA])[11], both in SJMM and O*NET data. For instance, the term "supporting" has a different meaning when used in the context of a responsibility role (e.g., "support our team in preparing reports") compared to its meaning as the core element of a work task (e.g., "supporting students with learning difficulties"). To assess potential benefit, we insert subspan markup tags into span texts and include them into the vocabulary for the LM and SBERT model tokenizers, too. This way, we ensure that embeddings are learned for each tag during MLM training and can be accessed during MNR training and at inference time. Table 2 provides examples

[10] Task refers here to the machine learning task of our model, not job tasks.

[11] Initial experiments incorporating different markup for every subspan type did not yield better results.

Query	Example
span only	*participate in reviews*
span only$_{MU}$	*[BOA] participate in reviews [EOA]*
span $\pm$1 span	develop QA instructions, *participate in reviews*, monitor audit measures
span $\pm$1 span$_{MU}$	develop QA instructions, *[BOA] participate in reviews [EOA]*, monitor audit measures
span + job title	*participate in reviews* as quality manager projects
span + job title$_{MU}$	*[BOA] participate in reviews [EOA]* as quality manager projects

Table 3: Examples of query configurations, target span in *italic* (examples translated from German to English).

of MNR training data processed in this manner. In general, we create two versions for each training data set presented above: one with plain text (e.g., T2Act) and one with subspan markup (e.g., T2Act$_{MU}$).

5.3. Query configuration

Individual job opening task spans can occasionally lack specificity or context. In such cases, the surrounding context, either from adjacent tasks or the job title, becomes crucial for a clear understanding. Therefore, we introduce multiple lookup configurations for evaluation. First, we include only the target span in the query. Second, we concatenate the target span with one adjacent task span to the left and one to the right, if available, separated by a comma. Third, we query the span in the form "task span as job title", closely resembling the training setting T*as*Occ2Act.

Models trained with markup are queried with markup as well. When there is contextual information surrounding the target span, we add the markup exclusively to the target span to highlight the specific area of interest. For a comprehensive illustration of our three query formulations, see Table 3.

5.4. Evaluation and Experiments

Evaluation Data: We devised a gold standard dataset consisting of 59 randomly sampled ads encompassing 312 task spans to assess our ontology mapping approaches. On average, each job advertisement in our evaluation dataset contains 5.3 task spans. For every single span, we collected the top three IWA mapping suggestions by various models and query settings, which human annotators then rated as 1 (highly appropriate), 0.5 (acceptable), or 0 (incorrect). To avoid bias towards specific models, we pooled the suggestions by different models and settings and presented them as a unified set for annotation. In total, about 8,900 IWA suggestions were annotated for the 312 task spans. Including the job title and neighboring tasks from the advertisement was crucial for providing contextual clues to the annotators, ensuring a reliable evaluation, particularly for shorter, ambiguous, or generic tasks (see Table 4).

A single annotator evaluated all 8k suggestions. Every span had at least one suggestion rated as acceptable (0.5). Over 98% of spans received at least one perfect suggestion. However, only 66% of the spans received three suggestions scoring a perfect 1. This dataset was then divided into a development set (30 ads with 169 spans) and a test set (29 ads with 143 spans) for further experimentation.

To measure inter-annotator agreement (IAA), four annotators independently evaluated a random subset of 10 spans, each with 10 IWA suggestions. The resulting Krippendorff's Alpha of 0.708 indicates satisfactory agreement among annotators but also reflects the inherent complexity of the mapping.

Experiments I: First, we assess the different approaches for learning a suitable vector space, in particular the various MNR pre-fine-tuning settings and the potential benefit of incorporating task subspan markup into training and inference. We also evaluate the impact of different contextualizations of queries. Performance metrics are measured on the development set and compared to a baseline performance achieved with a general domain SBERT model for German[12].

Experiments II: Second, we measure the effect of task-specific fine-tuning with IWA-labeled job ad task spans on a subset of high-performance configurations from experiments I.

Experiments III: Third, we choose models based on their performance in the development set, evaluating the advantages of model ensembling techniques on the test set, and establishing optimal similarity thresholds for real-world application.

Given the fine-grained classification involving over 330 IWA classes, conducting a thorough recall analysis is impractical. Thus, we prioritize the quality of the top IWA candidates and evaluate our suggestions using two main metrics. First, we calculate the accuracy of the top suggestion, counting it as correct only if it has a manual score of 1 (p@1). Second, we evaluate the quality of the top suggestions, up to a maximum of three (using a cutoff threshold determined by the similarity distribution

[12]Available on Hugging Face: gbert-large-paraphrase-cosine

[BOA] monitor legal compliance [EOA] [BCO] in environmental and conservation matters [ECO]		
IWA suggestion	A	B
[BOA] Evaluation of features or effects of regulations or policies [EOA].	0.5	1
[BOA] Give advice on environmental sustainability or environmentally friendly practices [EOA] to others.	0	0
[BOA] Evaluation of environmentally friendly technologies or processes [EOA].	0	0.5
[BOA] Evaluation of compliance with environmental standards or regulations [EOA].	1	1
[BOA] prepares and validates monthly, quarterly and annual financial statements [EOA].		
IWA suggestion	A	B
[BOA] estimate project development or operating costs [EOA].	0.5	0
[BOA] determination of values or prices [EOA] [BCO] for goods or services [ECO]	0	0
[BOA] Calculate financial data [EOA].	1	1
[BOA] Audit of financial activities, operations or systems [EOA].	0.5	0.5

Table 4: Examples of models' IWA suggestions for job ad tasks (in bold), assessed by annotator A and B. Examples are translated from German to English.

of 20 suggestions), in a more lenient manner. In this method, suggestions securing a score of 0.5 or above are considered correct but are weighted based on their respective scores (pW@c).

5.5. Results and Discussion

5.5.1. Experiments I

The results for the development set are presented in Table 5. The best-performing model and query configuration settings achieve a precision of 0.763 (p@1) and a weighted cutoff precision (pW@c) of 0.758. There is a considerable performance range across different settings, but all are much stronger than the baseline. This highlights the substantial genre shift between job ads and O*NET and the significant benefit of model adaptation to specific domains and tasks in our scenario.

In examining the **impact of training data settings**, we focus on the best results derived from various query and markup settings. Notably, the T*as*Occ2Act setting demonstrates an improvement of nearly three points in p@1 compared to T2Act, while pW@c exhibits nearly identical performance. In contrast, the T*or*Occ2Act setting does not yield any improvement over T2Act. This suggests that incorporating information on the occupational distribution of tasks in O*Net is beneficial. However, how this data is integrated into MNR training is crucial. Aligning MNR training pairs with hierarchical O*NET information and introducing occupations as contextual elements to task and activity pairs helps. On the other hand, introducing pairs of occupations and tasks or activities into the training data, thereby blending a different ontological relationship type into MNR pairs, seems to harm the adaptation.

The **query configuration** at inference time significantly influences outcomes, and its effects are closely intertwined with the data format provided during training. Consistency between training and querying at inference seems crucial. Notably, job titles in queries help when training data also includes job titles, as seen in the positive effects of this query in the T*as*Occ2Act setting (plus 3.0 points in p@1, compared to querying with span only), but they have negative effects in settings T2Act and T*or*Occ2Act, where no job title contextualization happened during training. Similarly, the effectiveness of span-only queries is maximized when training aligns with this approach, as evidenced in query comparisons within the T2Act setting. This emphasizes the importance of aligning training and query methods for optimal performance.

The role of markup is also intricately linked with training data settings. Markup yields a substantial improvement, particularly in T2Act (resulting in a 2.4-point increase in p@1 for span queries) and T*as*Occ2Act (where queries with job titles yield a 5-point increase in p@1). For T*as*Occ2Act, marking the boundaries of tasks and job titles clarifies the role of both components for IWA classification. The positive effect for T2Act further emphasizes the benefits of subspan markup, offering insights into the internal components of tasks, potentially clarifying the role of elements such as Responsibility Roles or Context. Conversely, in T*or*Occ2Act, the absence of markup proves more effective, indicating that in a learning setting that blends two distinct ontological relation types, this blending works better when there is also no separation through markup.

5.5.2. Experiments II

When assessing the **benefit of task-specific fine-tuning**, we focus on successful settings, namely T2Act and T*as*Occ2Act. MNR fine-tuning with labeled spans leads to a strong 10-point improvement, increasing the best model's p@1 to 0.763. Similarly, pW@c improved from 0.697 to 0.758, marking a 6.1-point increase. Hard negatives show more significant improvements, particularly in p@1.

In the T2Act setting, the update delivered a boost of at least five points for both query configurations and evaluation measures. Not surprisingly, span-only remained the best query configuration. Con-

	Query Configurations					
	Span		Span ±1 Span		Span + Job Title	
SBERT Models	p@1	pW@c	p@1	pW@c	p@1	pW@c
baseline	0.331	0.473	0.260	0.376	0.201	0.330
T2Act	0.615	0.663	0.367	0.437	0.515	0.610
T2Act$_{MU}$	0.639	**0.697**	0.367	0.443	0.533	0.609
+ *Ads*$_{MU}$	*0.692*	*0.739*			*0.586*	*0.680*
+ *AdsHN*$_{MU}$	*0.734*	*0.750*			*0.615*	*0.698*
T*as*Occ2Act	0.574	0.671	0.331	0.421	0.609	0.686
T*as*Occ2Act$_{MU}$	0.633	0.690	0.402	0.456	**0.663**	0.694
+ *Ads*$_{MU}$	*0.716*	***0.758***			*0.692*	*0.747*
+ *AdsHN*$_{MU}$	***0.763***	***0.758***			0.*704*	*0.747*
T*or*Occ2Act	0.615	0.678	0.272	0.358	0.533	0.592
T*or*Occ2Act$_{MU}$	0.574	0.644	0.231	0.356	0.509	0.586

Table 5: Performance of models and query configuration on dev set (n=169 spans). Best evaluation results before the task-specific fine-tuning in **bold**, and after fine-tuning in ***bold italic***.

versely, in the T*as*Occ2Act setting the span-only query emerged as the optimal selection, showcasing remarkable improvements of 10.0 points in p@1 and 6.8 points in pW@C. Through the update with labeled spans, this model seems to show a reduced dependency on job titles.

5.5.3. Experiments III

The **test set results** for the top-performing models, both before and after the update with job ad spans, are summarized in Table 6, showing average performance across five runs. We explore model ensembling by combining predictions from five models trained with the same configuration. Initial experiments on the development set revealed the effectiveness of blending ranks and similarities into the prediction score, with ranks receiving a weight of 3 and similarities a weight of 1 (see details in the appendix).

The test set performance, as reported in Table 6, exhibited a decrease of roughly five to ten points compared to the development set. This could be due to overfitting or due to the relatively modest size of datasets. Improvements by the job ad update were smaller on the test set than on the development set, with the largest gains being just over five points in p@1 and over seven points in pW@C, resulting in p@1 around 0.637 and pW@C just above 0.71. In general, the test set findings align with previously discussed trends.

Regarding **model ensembling over five runs**, improvements were observed across most settings and measures. For the best model, T2Act after the ads span update, ensembling brings a notable p@1 improvement of 2.0 points, reaching a score of 0.657, and a pW@C improvement of 3.4 points, achieving a score of 0.752.

Finally, we conducted an assessment of **ensembling different models and query settings**, employing the three best development set configurations: T2Act after the update with ad spans, queried with the span-only (B), and T*as*Occ2Act after the update, queried with both span-only (D) and span plus job title (E). Ensembles over ten runs (five runs of two configurations each) produced significant performance improvements. However, ensembling over 15 runs (five runs of three configurations each) did not yield further gains. Notably, ensembling T2Act queried with span-only, and T*as*Occ2Act queried with job titles (B, E), boosted p@1 to 0.699, a more than 4-point improvement over single-model ensembling. This hints at the benefits of ensembling different query and training settings.

Model	**Query**	p@1	pW@c
T2Act$_{MU}$	span	0.584	0.678
5x-ensemble (A)	span	0.587	0.713
+ AdsHN$_{MU}$	span	0.637	0.718
5x-ensemble (B)	span	**0.657**	**0.752**
T*as*Occ2Act$_{MU}$	span + job title	0.597	0.696
5x-ensemble (C)	span + job title	0.601	0.751
+ AdsHN$_{MU}$	span	0.637	0.716
5x-ensemble (D)	span	0.650	0.738
+ AdsHN$_{MU}$	span + job title	0.623	0.713
5x-ensemble (E)	span + job title	0.622	0.733
10x-ensemble mixed (B, D)		0.671	***0.761***
10x-ensemble mixed (B, E)		***0.699***	0.755
10x-ensemble mixed (D, E)		0.678	0.740
15x-ensemble mixed (B, D, E)		0.643	0.651

Table 6: Mean and ensemble performance of models on test set (n=143 tasks), model selection on dev set performance. Best evaluation measure for single configurations in **bold**, for mixed ensembles in ***bold italic***.

For the best ensemble setting, we assessed the consequences of excluding subspan markup during inference. The markup removal had a clear adverse impact, leading to a 5.6-point loss in p@1. This

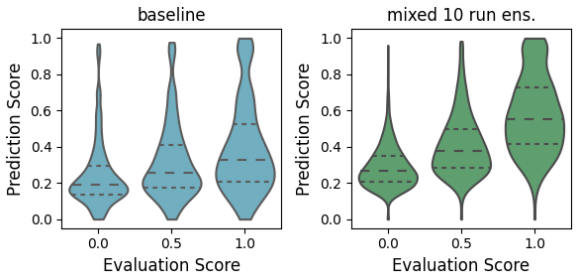

Figure 2: Normalized prediction score distribution by evaluation score, on test set, for baseline and best mixed ensemble.

underscores the importance of preserving subspan markup during inference to sustain performance.

Figure 2 presents the **distribution of prediction scores** for perfect, acceptable, and incorrect IWA suggestions, comparing the best ensembling model to the baseline. The best model generally exhibits higher prediction scores, suggesting a reduced semantic gap between ontology and job ad data. Notably, the interquartile range for perfect suggestions is wider than for incorrect ones, indicating that when there are multiple perfect IWA suggestions for a query, models do not consistently provide high ranks or similarities for all of them. But most importantly, the best model shows a more substantial difference in prediction scores between correct and incorrect IWA suggestions compared to the baseline. For instance, the median value difference between evaluation scores 1 and 0 is 0.286 for the best model (0.55 vs. 0.26), while only 0.136 for the baseline (0.327 vs. 0.191). Moreover, it can be observed that 75% of perfect solutions produced by the best model have values exceeding 0.4, while over 75% of incorrect suggestions fall below this threshold, highlighting the utility of this model's prediction scores.

Building upon this, for practical application, we aim to determine an **optimal similarity threshold** for filtering out hits without suitable IWA candidates, either due to erroneous extractions or because certain tasks are not represented in the U.S.-oriented O*NET. We experimented with a range of thresholds on the development set and selected one that yielded an improved p@1 while excluding only a handful of cases. When applied to the test set, this threshold excludes two cases and results in p@1 of 0.709. In a more lenient setting, considering evaluation scores of 0.5 as correct too, we reach p@1 of 0.908.

6. Conclusion

This work addresses the challenge of extracting specific job tasks from German job postings and mapping them to fine-grained work activities in the O*NET labor market ontology, resembling a fine-grained multi-label classification problem. We employed SBERT models to enable semantic similarity comparisons between job ad tasks and O*NET work activities, utilizing the available ontological data with a Multiple Negatives Ranking loss, and integrating a limited amount of labeled job ad data into the training process. The results were promising, with our best model achieving a p@1 of approximately 70% on the test set, marking a significant improvement compared to a baseline of 33% by a general-domain SBERT.

The integration of occupational information from O*NET, especially when leveraging occupational titles for contextualizing hierarchical job task relationships with work activities, proved to be of great importance in our experimentation with training data.

However, updating our models with a relatively small amount of labeled job ad data, focusing on cases that proved difficult for pre-fine-tuned models, brought the most substantial performance improvement. This approach addressed the need to bridge the semantic gap between ontology and job ad data in the most direct manner.

Our experiments demonstrated the value of incorporating subspan markup during training and inference. Markup likely supported accurate classification by clarifying the role of subspans in a task and by streamlining the format between diverse formulations in job postings and more structured, uniform ontological work activities. We further showed, that aligning query formulations with the data format provided in training yielded favorable results.

For practical use, we investigated model ensembling, which notably enhanced performance, particularly when combining various training and query configurations. We additionally set an effective filtering threshold to exclude inappropriate IWA suggestions. This led to strong performance, with highly appropriate first-ranked IWA suggestions in 7 out of 10 cases and acceptable IWA suggestions in 9 out of 10 cases, despite the task's inherent complexity.

Our work opens up numerous opportunities for social science research, such as exploring the combination of different tasks into jobs and occupations, as well as tracking their evolution over time, to highlight only the most evident examples. Additionally, the successful mapping of tasks from Swiss job postings to work activities in the US labor market ontology O*NET, effectively bridges the gap between two languages and two labor markets, allowing for future international comparisons.

7. Acknowledgements

This work is supported by the Swiss National Science Foundation (grant number 407740 187333). We thank the anonymous reviewers for their careful reading of this article and their helpful comments and suggestions, and Jan Müller and Yanik Kipfer for their efforts in post-evaluation.

8. Bibliographical References

Daron Acemoglu, David Autor, Jonathon Hazell, and Pascual Restrepo. 2022. Artificial intelligence and jobs: evidence from online vacancies. *Journal of Labor Economics*, 40(S1):S293–S340.

Enghin Atalay, Phai Phongthiengtham, Sebastian Sotelo, and Daniel Tannenbaum. 2020. The Evolution of Work in the United States. *American Economic Journal: Applied Economics*, 12(2):1–34.

Marlis Buchmann, Helen Buchs, Eva Bühlmann, Ann-Sophie Gnehm, Debra Hevenstone, Yanik Kipfer, Urs Klarer, Jan Müller, Marianne Müller, Stefan Sacchi, Alexander Salvisberg, and Anna Von Ow. 2023. Swiss job market monitor 1950-2022 (8.0.0) [dataset].

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

Jens-Joris Decorte, Jeroen Van Hautte, Johannes Deleu, Chris Develder, and Thomas Demeester. 2022. Design of negative sampling strategies for distantly supervised skill extraction.

Jens-Joris Decorte, Severine Verlinden, Jeroen Van Hautte, Johannes Deleu, Chris Develder, and Thomas Demeester. 2023. Extreme multi-label skill extraction training using large language models. *arXiv preprint arXiv:2307.10778*.

David Deming and Lisa B. Kahn. 2018. Skill requirements across firms and labor markets: Evidence from job postings for professionals. *Journal of Labor Economics*, 36(S1):S337–S369. Publisher: University of Chicago Press Chicago, IL.

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

Ann-Sophie Gnehm, Eva Bühlmann, Helen Buchs, and Simon Clematide. 2022. Fine-grained extraction and classification of skill requirements in German-speaking job ads. In *Proceedings of the Fifth Workshop on Natural Language Processing and Computational Social Science (NLP+CSS)*, pages 14–24, Abu Dhabi, UAE. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. 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.

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. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Mike Zhang, Kristian Nørgaard Jensen, and Barbara Plank. 2022. Kompetencer: Fine-grained skill classification in Danish job postings via distant supervision and transfer learning. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 436–447, Marseille, France. European Language Resources Association.

Mike Zhang, Rob Van Der Goot, and Barbara Plank. 2023. ESCOXLM-R: Multilingual Taxonomy-

driven Pre-training for the Job Market Domain. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11871–11890, Toronto, Canada. Association for Computational Linguistics.

Appendices

A.1 MLM training parameters

We use the Hugging Face Transformers library (Wolf et al., 2020) to conduct continued in-domain pretraining of the transformer-based JobGBERT model[13], largely adhering to their default settings for MLM. Parameter settings remain consistent for both training on plain text and training on text with subspan markup. Subspan markup tags are incorporated into the vocabulary for model tokenizers. Our approach includes a maximum sequence length of 512 subwords and initiates training with a learning rate of 5e-5 using a linear learning rate schedule with a warm-up ratio of 0.05. We use the Adam optimizer with β_1 of 0.9 and β_2 of 0.999, and ϵ of 1e-8. Training occurs over 3 epochs, utilizing a batch size of 256 on 2.2 million text spans, resulting in a total of 25,800 training steps. Training happens on an NVIDIA Tesla T4 with 16 GB of RAM for approximately 10 days.

A.2 MNR training parameters

MNR training utilizes the Sentence-BERT library (Reimers and Gurevych, 2019). We employ the [CLS] token's output embedding as the sentence representation and cosine similarity with a scaling factor of 50 for measuring sentence similarity. Training incorporates a linear rate scheduler with a learning rate of 2e-5 and a minimal warm up of 10 steps. The Adam optimizer is utilized with a weight decay of 0.01, β_1 of 0.9, β_2 of 0.999, and ϵ of 1e-6. Batch size is set to 32. For comparability, in all settings of Experiment I, we train for 138,550 steps (corresponding to 50 epochs for T2Act, roughly 10 epochs for T*as*Occ2Act, and 25 epochs for T*or*Occc2Act), and in Experiment II, training lasts for 2640 steps (10 epochs). Training happens on an NVIDIA Tesla T4 with 16 GB of RAM.

A.3 Prediction score in model ensembles

The prediction score for combining candidate suggestions by different models in ensembling is given in Equation 1, where *r* is the rank of a suggestion by a model and w_r is the weight for ranks (3 in our case), *s* is the cosine similarity provided by a model, w_s is the weight for cosine similarities (1 in our case), and n is the number of models in the ensemble (5, 10, or 15 in our case).

$$\mathrm{y} = \sum_{i=1}^{n} \left(\mathrm{s}_i \times \mathrm{w}_s + \mathrm{w}_r \times \sqrt{\frac{1}{\mathrm{r}_i}} \right) \quad (1)$$

We considered the top 20 suggestions per model for ensembling. If a candidate from one model's top 20 was not among the top 20 of another model, we assigned a default rank value of 21 and a similarity value of the 20th candidate minus 0.00001.

[13] Available on Hugging Face: jobGBERT