Enhancing PLM Performance on Labour Market Tasks via Instruction-based Finetuning and Prompt-tuning with Rules

Jarno Vrolijk[1], David Graus[2]

[1]*University of Amsterdam, Amsterdam, The Netherlands*
[2]*Randstad, Diemen, The Netherlands*

Abstract

The increased digitization of the labour market has given researchers, educators, and companies the means to analyze and better understand the labour market. However, labour market resources, although available in high volumes, tend to be unstructured, and as such, research towards methodologies for the identification, linking, and extraction of entities becomes more and more important. Against the backdrop of this quest for better labour market representations, resource constraints and the unavailability of large-scale annotated data cause a reliance on human domain experts. We demonstrate the effectiveness of prompt-based tuning of pre-trained language models (PLM) in labour market specific applications. Our results indicate that cost-efficient methods such as PTR and instruction tuning without exemplars can significantly increase the performance of PLMs on downstream labour market applications without introducing additional model layers, manual annotations, and data augmentation.

Keywords

taxonomy, transformer, natural language processing, labour market intelligence

1. Introduction

The increasing availability of raw labour market information allows businesses, educational facilities and job seekers to gain a clear and more complete understanding of the labour market [1]. While the increasing volumes of available data provide opportunities, there are several challenges towards fully utilizing the data.

On the one hand, the majority of available data is of an unstructured nature, with a lack of large-scale annotated datasets that could be used in training, and/or fine tuning of models for downstream applications. On the other hand, there is much effort in creating structured representations of labour market data, through taxonomies and ontologies such as ESCO,[1] ISCO,[2] or O*NET.[3]

Leveraging structured ontologies to enrich and interpret unstructured labour market data has considerable research attention, through skill and occupation recognition, classification, and linking [2, 3, 4, 5, 6]. These different downstream tasks can prove invaluable in enabling better workforce and labour market insights, identification of trends and temporal patterns, and providing structured data or enrichments that can be applied as feature representation for job or career path recommen-

dations [7, 8, 9].

Many of these approaches rely on supervised learning, where a commonly identified limitation in literature is the availability of multilingual, labour market and task-specific datasets. In addition, due to the dynamic nature of the labour market makes it very difficult to keep more structured representations of the labour market up-to-date and relevant (i.e. labour market ontologies and taxonomies); updating and maintaining these knowledge structures is typically done by human domain experts, making them time- and resource-intensive, and meaning that whenever such a structure is updated, datasets for supervised learning may become obsolete.

In this paper, we propose a novel method that relies on pretrained language models (PLMs), prompt tuning with rules (PTR) [10], and the structured multilingual ESCO taxonomy, to efficiently and cheaply generate large amounts of labeled data for learning a variety of downstream tasks for extracting structured information from unstructured labour market data, specifically: (i) relation classifiers, that aim to predict the type of relation between skills and occupations, (ii) entity classifiers, that aim to classify labour market entities as skill or occupation, (iii) entity linkers, which aims to link various surface forms of labour market entities to their canonical underlying skill or occupation entity, and (iv) question answering approaches, that aim to answer the correspondence between a descriptive text and the associated skill or occupation.

In this paper, we aim to address the following research questions:

1. Are "out-of-the-box" PLMs capable of generalizing learned behavior to labour market specific

[1]https://esco.ec.europa.eu/en
[2]https://www.ilo.org/public/english/bureau/stat/isco/isco88/
[3]https://www.onetcenter.org/overview.html

applications?

2. Does instruction, and sub-prompt finetuning a PLM on a mixture of task-specific (i.e. general and labour market specific) datasets increase the performance on labour market specific benchmarks?

3. Is the tuned PLM able to transfer the learned behavior across labour market specific tasks?

In this paper, we demonstrate domain-specific prompt-based tuning and its effect on the performance of skill extraction, occupation classification, link prediction, and entity linking tasks. We propose leveraging instruction tuning without exemplars (i.e. no examples at inference time) and sub-prompts for a more cost-efficient solution for downstream labour market applications [11, 12, 13, 10]. We provide manually constructed templates that encode the knowledge embedded in the ESCO occupation and skill taxonomies. We benchmark different configurations of finetuning the PLMs, to demonstrate the effectiveness of e.g., adding instructions or sub-prompts.

2. Related Work

Recent successes of PLMs such as GPT [14], BERT [15], RoBERTA [16] and T5 [17] have demonstrated the usefullness and adaptability of the transformer architecture. Although these PLMs can capture rich knowledge from massive corpora, a fine-tuning process with extra task-specific data is still required to transfer their knowledge for downstream tasks. Besides fine-tuning language models for specific tasks, recent studies have explored better optimization and regularization techniques to improve fine-tuning.

Several works try to integrate ontological and/ or taxonomical knowledge into task-specific models, to improve the performance of downstream applications. Take the work by [18], who introduced *KnowBert*, a methodology that explicitly models entity spans in the input text. They further use an entity linker to retrieve relevant embeddings of the entity from a knowledge base to enhance their representations. Another approach would be the work by [19], the so-called *TransE* model, that focused primarily on representing hierarchical relationships. Similar to our work, *TransE* models multi-relational data from knowledge bases (i.e. triplestores) to improve performance for link prediction [19].

[20] took a different approach, proposing *ERNIE*, a method that consists of two stacked modules, namely the T-Encoder responsible for capturing lexical and syntactic information, and the K-Encoder responsible for augmenting this lexical and syntactical information with extra token-oriented knowledge from the underlying layer [20]. Lastly, we have *ESCOXLM-R* that employ further pre-training on the ESCO taxonomy [6]. In addition to

the masked language modelling (MLM) pre-training objective, the authors also introduce the so-called ESCO Relation Prediction (ERP) task to internalize knowledge of non-hierarchical relations within ESCO [6].

Another pre-training-based approach that leverages a self-supervised method to pre-train a deeply joint language-knowledge foundation model from text and knowledge graphs at scale is the Deep Bidirectional Language-Knowledge Graph Pretraining (DRAGON) proposed by Yasunaga et al. [21]. Results from the paper indicate that DRAGON outperforms existing LM and LM+KG models on diverse downstream tasks in particular on complex reasoning about language and knowledge.

Despite the success of fine-tuning PLMs, there is a big gap between the MLM objective and fine-tuning objectives for downstream applications. Prompt-based learning has been a widely explored method that uses templates to transform the input into classification problems, and as such, closes the gap between task-specific and MLM objectives [22]. [23] propose *KnowPrompt*, a method for task-oriented prompt template construction where they use special markers to highlight entity mentions in the template. [24] also proposed a template-based NER model using BART. The model enumerates all possible text spans and considers the generation probability of each type within manually crafted templates [22, 24].

Since the manual creation of templates is labour-intensive, methods for the automated generation of prompts and labels are well-researched. In principle, a prompt consists of a template and label words. As such, Schick and Schütze [13] first searches the label word space for the manually created templates. Next, gradient-guided search automatically generates both templates and label words. Compared to human-picked prompts, most auto-generated prompts cannot achieve comparable performance [10].

Prior literature has shown that increasing the number of tasks in finetuning improves the generalization to unseen tasks [11]. Experiments from Chung et al. [11] show that "instruction finetuning" scales well with the number of tasks and the size of the model. Wei et al. [25] further suggests that instruction-tuned models respond better to continuous outputs from prompt tuning. Prompt tuning on FLAN even achieves more than 10% improvement over a non-instruction-tuned equivalent model [25].

3. Methodology

3.1. Preliminaries

3.1.1. ESCO

ESCO (European Skills, Competences, Qualifications and Occupations) is the European multilingual classification
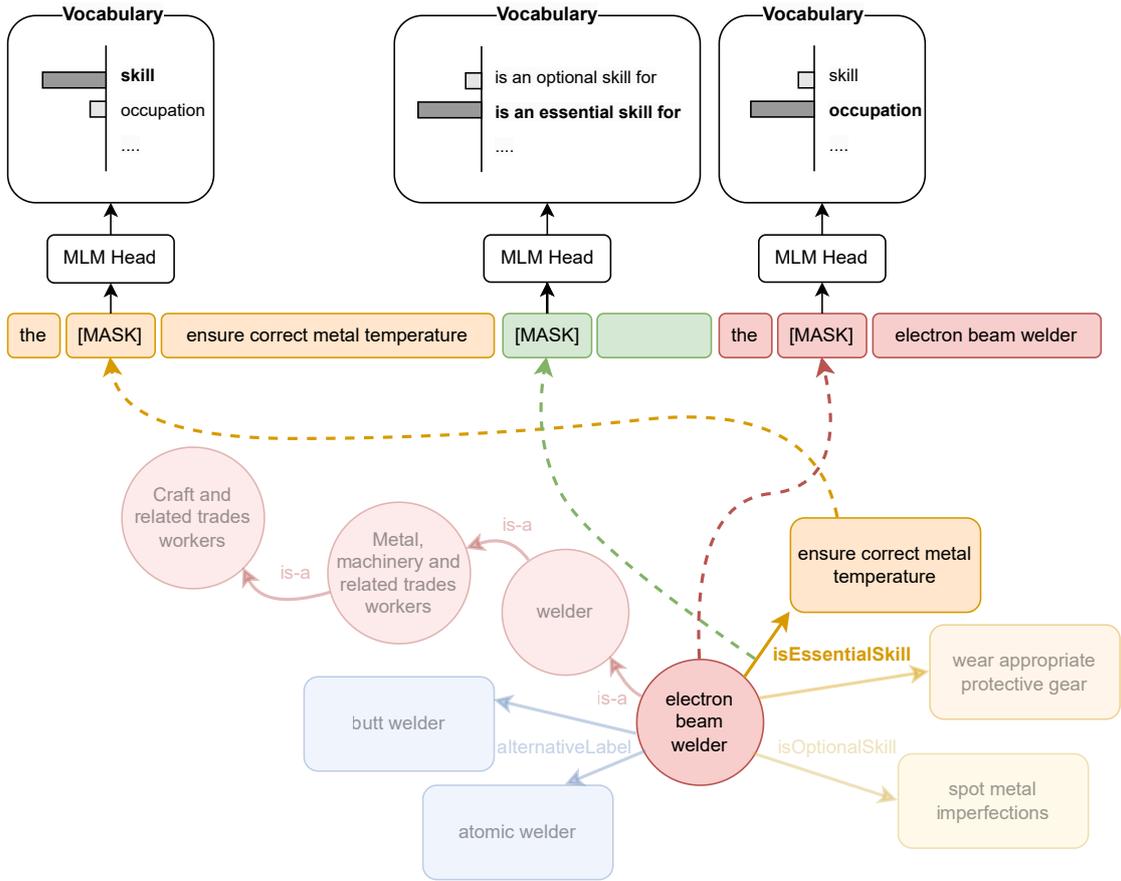
Figure 1: Visual representation of proposed method: PTR (shown on top) with three sub-prompts (yellow, green, and red) with MLM heads predicting [MASK] tokens, given their respective verbalizers (inspired by Han et al. [10]). An outtake of the ESCO taxonomy represented in the bottom, with hierarchical (red) relations and non-hierarchical (rest), and how the entities and relations populate the template (dotted lines).

of skills, competences and occupations. In total, ESCO describes 3,008 occupations and 13,980 knowledge, skill, and competences in 28 different languages.

ESCO has both hierarchical and non-hierarchical relationships: hierarchical relationships, or hypernymies, are relations of the form *x is-a y* [26, 4]. Non-hierarchical relationship are essentially those relationship that are not hierarchical. For example: "*Java Programming* is an essential skill for a *Software Developer*" is a non-hierarchical relationship, whereas, "a *Software Developer* is a *Information and Communications Technology Professional*" is hierarchical.

3.1.2. PLMs: T5 & FLAN-T5

In this paper we rely on the T5 PLM, since the text-to-text framework allows us to directly apply the same model, objective, training procedure, and decoding process to every task we consider [17]. In addition, we turn to an instruction-tuned variant of the T5 model: FLAN-T5. Instruction-based finetuning has shown to improve zeroshot performance on unseen tasks [25, 11]. In this paper we aim to study whether this property also applies to the domain-specific unseen tasks that we propose.

3.2. Prompt-Tuning with Rules

In this paper, we utilize the ESCO taxonomy as background knowledge for the Prompt-Tuning with Rules (PTR) approach proposed by Han et al. [10].

PTR builds on prompt tuning methods that rely on cloze tests, where the PLM is applied to replace or fill in a missing word in a sentence. A so-called verbalizer maps a fixed set of class labels (e.g., `positive`, `negative`) to underlying label words (e.g., "great", "terrible"), so that

by predicting a label word, the PLM effectively classifies a sentence.

PTR extends this prompt tuning approach with prior knowledge encoding, i.e., leveraging logic rules to encode prior knowledge about tasks and classes into prompt tuning, and efficient prompt design, through composing multiple sub-prompts and combining into prompts [10].

Illustrative example   We illustrate how we leverage the ESCO taxonomy to construct and populate sub-prompts, as proposed by Han et al. [10].

Consider a (sub-)prompt template for entity type classification, as; `"[CLS] the [MASK] [ENTITY]."` Which can be instantiated for the skill "ensure correct metal temperature," as: `"[CLS] the [MASK] ensure correct metal temperature,"` and for the occupation "electron beam welder," as: `"[CLS] the [MASK] electron beam welder."`

Finally, we can combine the above instantiations of the same sub-prompt into a final prompt, that spans entity type and relation classification, as such: `"[CLS] the [MASK]₁ ensure correct metal temperature [MASK]₂ the [MASK]₃ electron beam welder"`.

PTR relies on so-called "verbalizers" that map class labels to label words. In our example, the class labels {"skill", "occupation"} for entity classification are mapped to (the same) label words {"skill", "occupation"} in place of [MASK]$_1$ and [MASK]$_3$, and the class label {"isEssentialSkill", "isOptionalSkill"} in place of [MASK]$_2$ are mapped to the corresponding label words {"is an essential skill for", "is an optional skill for"} in the case of relation classification.

i.e., $\varphi_{[MASK]_1}$ and $\varphi_{[MASK]_3}$ aim to assign an entity class through predicting a label word from X, and $\varphi_{[MASK]_2}$ aims to classify the type of relation between the two through label words Y.

3.3. Instruction-based Finetuning

Instruction-based finetuning aims to teach a PLM to perform certain tasks, by responding to instructions in natural language [25]. For two of our three datasets (i.e., the QA and EL), we manually constructed templates that result in natural language instructions that describe the task for that dataset to the PLM.

While scaling language model sizes seems to be a reliable predictor for improved model performance, it comes at the price of high compute. Therefore, development of compute-efficient techniques that improve performance at the cost of a relatively small amount of computational resources is important. Instruction-based finetuning improves performance of PLMs on evaluation benchmarks by up to 9.4%, requiring only 0.2% of the pre-training

Figure 2: Visual representation of method: Instruction tuning for the QA examples. The instruction is prepended to the question, instructing the PLM how to proceed in answering the given question.

compute [11]. Furthermore, Chung et al. [11] demonstrate that smaller models that are instruction tuned can outperform larger models without it.

Figure 2 demonstrates how we leverage the ESCO taxonomy to construct instruction tuning templates for the QA examples.

4. Experimental Setup

The aim of this paper is to leverage prompt-based and instruction-based finetuning, to cost-efficiently optimize PLMs performance on four downstream labour market tasks. As described in the previous section, we propose four different tasks for evaluation, namely: entity classification (EC), relation classification (RC), entity linking (EL), and question answering (QA).

4.1. Datasets

We evaluate PTR and instruction-based finetuning in labour market-specific downstream tasks through benchmark datasets we generate through populating handcrafted templates, with instances from the ESCO taxonomy.

We generate three datasets of prompts, that address four different tasks; i) entity classification (EC) and ii) relation classification (RC) as illustrated above (combined in a single set of prompts), in addition to iii) entity linking (EL), and iv) question answering (QA).

Construction of a self-supervised dataset comprises three different components; i) a subset of ESCO relations, ii) a template to map the triples associated to the relations

	EC + RC	QA	EL
# total	123,752	27,792	195,350
# skills	13,890	13,890	13,890
# occupations	3,008	3,008	3,008
# essential	64,877	-	-
# optional	58,875	-	-
# altlabels	-	-	96,117
# pos	123,752	13,896	97,675
# neg	0	13,896	97,675

Table 1
Statistics of the different datasets. Since, the train and evaluation sets differ due to random sampling or the choice for K, we can only report the total counts.

to (sub-)prompts, and iii) verbalizers that map class labels to label words.

4.1.1. Entity Classification + Relation Classification

To build the entity classification and relation classification (EC + RC) dataset, we leverage the *isEssentialFor* and the *isOptionalFor* relations as found in ESCO. For both entity classification and relation classification, we largely follow the work by Han et al. [10], i.e., we extract all triples that have as subject a skill entity, the *isEssentialFor* or the *isOptionalFor* as predicate, and finally as object an occupation entity.

Formally, our triples look as follows:

$$< Skill, r, Occupation >, \tag{1}$$

where $r \in \{isOptionalFor, isEssentialFor\}$. The entity and relation classification template $T(x)$ is formalized as:

$$s_1 = The \ [MASK] \ entity \ [Skill]$$
$$s_2 = The \ [MASK] \ entity \ [Occupation]$$
$$s_1 \ [MASK] \ s_2$$

Lastly, we formulate two different verbalizers φ_1 and φ_2 such that:

$$\varphi_1 = C_1 \rightarrow \nu_1, \tag{2}$$
$$\varphi_2 = C_2 \rightarrow \nu_2, \tag{3}$$

where C_1={*'Occupation', 'Skill'*} and the accompanying label words ν_1 = {*'occupation', 'skill'*}. Similarly, C_2 = {*'isOptionalFor', 'isEssentialFor'*}, and the label words ν_2 = {*'is optional for', 'is essential for'*}.

Note that in our case, the verbalizers are one-to-one mappings, whereas in the PTR methodology, many-to-one mappings are also supported. For the entity and

relation classifications we have not included the possibility of "no relation" and/ or "no entity", for the simple reason of self-supervision. While we fully believe these negative examples to be useful for better learning how to recognize entities and the relation connecting entities, they would require manual annotation, and as such fall beyond the scope of this research.

4.1.2. Entity Linking

To model entity linking, we rely on the *alternativeLabel* relation in ESCO, i.e., our task is to map an entity surface form or entity mention (*alternative label*), to the canonical entity name (i.e., *label*).

We can formalize the entity linking task as the following triplestore:

$$< e, r, m >, \tag{4}$$

where $e \in E$ the set of skill and occupation entities, and $m \in M$ the set of skill and occupation mentions (i.e., alternative labels for the ESCO skill and occupation labels). Lastly, $r \in C$, meaning that the predicate can be either signalling that the mention is an alternative label or not an alternative for the given ESCO entity.

Given an entity, e and a mention m, we are interested in finding out what type of relation there exists between e and m. As such, we formalize the entity linking as a masked language problem via $x_{prompt} = e \ [MASK] \ m$.

In Figure 1, the blue boxes represent two examples of alternative labels for the occupation *electron beam welder*.

We formalize the template for our second set of prompts as:

$$e \ [\text{MASK}] \ m$$

We formulate the verbalizer φ such that:

$$\varphi = C \rightarrow \nu, \tag{5}$$

where

$$C = \{\texttt{alternativeLabel}, \texttt{noAlternativeLabel}\}$$
$$\nu = \{\texttt{is an synonym for}, \texttt{it not a synonym for}\}$$

For each generated example from the ESCO triplestores, we also randomly sample negative examples by randomly shuffling the objects and subject of the positive triplestores and changing the predicate label to *noAlternativeLabel*.

4.1.3. Question Answering

For the QA task, we use so-called *instructional templates* as defined by Chung et al. [11] and Wei et al. [25]. Instructional templates prepend an instruction to the prompt.

In our case, we prepend the example and question with *"Answer the following with yes/no"*, instructing the PLM how to answer the question that follows.

The question and answering dataset is constructed with the descriptions of the entities in ESCO. As such, we can construct a dataset as:

$$< e, \{\texttt{description}\} > \qquad (6)$$

Next, we define the template $T(x)$ as depicted in the example in Figure 2. Where we first prepend the instruction *"Q: Answer the following with yes/no"* to the body *"Does [description] describe [entity label]"*, to finish it off with *"A: [MASK]"*.

The verbalizer φ then maps the $\{'yes','no'\}$ to the label words $\{'yes','no'\}$.

We randomly sample correct examples, in addition to generating negative examples by randomly sampling a skill or occupation entity, and pairing this with a randomly sampled description from the set of available descriptions, and tagging the label for the answer to be *"no"*. This results in a balanced dataset, with a fifty-fifty split of positive and negative examples.

4.2. Experiments

In order to answer our research questions, we propose the following experiments.

4.2.1. Experiment 1: Zero-shot Learning

First, to better understand the labour market-specific tasks that we propose, we first test off-the-shelve PLMs in a zero-shot setting, using our own generated prompt datasets for inference.

In addition, to test the hypothesis that FLAN-T5's multitask learning enables a better ability of learning additional (domain-specific) tasks, in our first experiment we directly compare off-the-shelve T5 and FLAN-T5 models, on each of our three datasets.

4.2.2. Experiment 2: K-shot Learning

Next, having established the performance differences between the off-the-shelve PLMs, we study the impact of few-shot learning to steer the best performing PLM from experiment 1 towards the domain-specific data and tasks, where we perform an ablation study on the number of examples (K) we use for few-shot learning.

This is motivated by a.o., Han et al. [10], who report comparable or even better results in the few-shot scenario than e.g., methods that inject special symbols to index the positions of entities and methods that inject both type information and special symbols. The authors sample K training instances and K validation instances per class

from the original training set and development set, and evaluate the models on the original test set.

We propose using $K = \{64, 128, 256\}$ sets.

4.2.3. Experiment 3: Multitask Learning

After having studied the effect of few-shot learning, we perform an ablation study to measure the effect of learning multiple tasks in parallel, i.e., transfer learning from one task to the other.

We do so by fine-tuning the FLAN-T5 on all combinations of tasks from a single to the full set, i.e., we train FLAN-T5 on the RC+EC and consecutively on the EL and QA tasks. We then test the performance of the resulting model on all three data sets to identify whether, e.g., prompt tuning on EL can help performance on the QA dataset.

4.3. Implementation Details

Our model implementation relies on the HuggingFace, PyTorch and OpenPrompt frameworks (albeit with some customizations), proposed by Wolf et al. [27], Paszke et al. [28] and Ding et al. [29] respectively.

For the zero-shot approach of the first experiment, we turn to T5 and FLAN-T5, for which we use the implementation by the original authors [17, 25, 11]. More specifically, we use the 3 billion parameter checkpoints as found on huggingface under the names; *'t5-3b'*, and *'google/flan-t5-xl'*.

To answer our second research question we adjust the number of examples used for training the models by comparing different values for parameter K (i.e., number of samples). We optimize our PTR and Instruction-based finetuning models using AdamW, with a learning rate of respectively $3e-5$ and $2e-5$. Furthermore, we reset the weight decay on the normalization layers and bias. We fine-tune all models using batch size 32, and train the PTR models for 10 epochs, whereas, we train the instruction based finetuning models for only 5. The best model checkpoint is selected.

4.4. Evaluation metrics

In order to systematically evaluate few-shot performance, we randomly pick K samples from the total dataset, and use the remaining data to sample evaluation sets. This sampling is done 9 times, each iteration we sample 512 random examples from the remaining data after the train/test split. We report F1 scores averaged over 9 runs in addition to standard deviations ($^{\pm std}$). We argue that sampling multiple splits gives a more robust measure of the actual performance.

Since the single EC+RC dataset contains two separate tasks, it is important to avoid contamination between the

Model	EC+RC	QA	EL
T5	$48.07^{\pm.19}$	$33.75^{\pm.2}$	$33.89^{\pm.37}$
FLAN-T5	$44.54^{\pm.66}$	$83.44^{\pm.44}$	$57.38^{\pm.60}$

Table 2
F1 scores of experiment 1, where we compare 0-shot performance between T5 and FLAN-T5.

train and test sets. Therefore, after the initial division, we check all individual skill and occupation entities from the train set, and remove all relations in the test set that contain any of those entities. For the QA and EL training data the risk of contamination is mitigated through the train/test split (i.e., after the normal split unique entries belong either to the train or test set).

5. Results

In this section we present and summarize the results of our experiments described in Section 4.2.

5.1. Experiment 1: Zero-shot learning

See Table 2 for the comparison of T5 and FLAN-T5 in 0-shot learning, i.e., applied off the shelve for inference on our generated prompts.

First, we see that FLAN-T5 substantially outperforms the non instruction-based finetuned counterpart T5 on the QA (83.44 vs. 33.75 respectively) and EL tasks (57.38 vs. 33.89 respectively), but slightly underperforms on the EC+RC tasks, at 44.54 for FLAN-T5 and 48.07 for the T5 model.

A potential explanation for this might be the fact that FLAN-T5 is trained on a variety of entity classification tasks that do not involve skill and occupation entities (i.e., the primary focus is on person and organisation entities). As such, the learned patterns may interfere with the PLMs ability to recognize skills and occupations.

5.2. Experiment 2: K-shot learning

In Table 3 we show the performance differences at different levels of k in the fewshot learning scenario.

First, we note how the performance of FLAN-T5 + PTR substantially outperforms both T5 and FLAN-T5 from Table 2 with F1 scores between 50.42 and 51.60 across different values of K, compared to 48.07 and 44.54 respectively for the zero-shot T5 and FLAN-T5.

Next, we see that different values of K are optimal for different tasks; with maximum scores at $K = 128$ for EC+RC and QA at 51.60 and 94.23 respectively, and a maximum score of 98.06 for $K = 256$ for EL.

The scaling of the model potentially gives us insights into how sample efficient the model is in learning the behavior. Larger models are in-general more sample efficient and as such require less examples to learn a particular behavior [30].

5.3. Experiment 3: Multitask learning

Finally, we show the impact of learning single or multiple tasks at once. Results of our ablation experiments are shown in Table 4, where we vary with models that are trained on all combinations of different train sets of prompts, which we evaluate on each of the three test set of prompts.

Here, we note that first, in some cases adding prompts for additional tasks increases performance for the original tasks, consider, e.g., the case for (testing on) EL, where adding QA prompts yields an F1-score of 97.61 (row 4, Table 4), and adding EC+RC prompts even gets performance up to 98.48 (row 5, Table 4), whereas the model tuned with EL prompts only, scores 95.17 F1 (row 3, Table 4).

However, this does not apply for QA nor EC+RC, where only tuning with respectively QA and EC+RC prompts yields the highest score, nor for the case of training on all additional prompts—these runs (bottom row in Table 4) do not outperform the best performing models tuned on one or two sets of prompts.

Overall, this indicates that multitask learning can contribute in some cases to increased performance.

5.3.1. Unseen task performance

Supporting these observations is the pattern around performance on unseen tasks, i.e., models tuned on (a) task(s) that do not include the test task used for evaluation. Consider, e.g., EL; models that have not seen any EL prompts in their tuning stage, perform substantially worse with 58.96 for EC+RC and QA, 57.40 for EC+RC, and 60.31 for QA, versus between 95.17 and 98.48 for models that have seen EL prompts.

Similar patterns are seen with the other tasks, where for EC+RC models that have not seen any EC+RC prompts perform between 45.05–47.68, and around 50.55 and 51.60 for models that have. For QA, we see that models without QA prompts in tuning score between 68.22–87.98, and models that have range from 93.24 to 94.23.

However, increasing the number of tasks in tuning does increase performance for unseen tasks in two out of three cases: when testing on the EC+RC prompts, a model that combines QA and EL prompts in tuning scores 47.68, and outperforms QA-only (45.93) and EL-only (45.04) models. Similarly, for QA, combining EC+RC and EL prompts yields an F1-score of 87.98, whereas EC+RC-only yields 78.36, and EL-only a mere 68.22 F1.

Model ↓/K →	EC+RC			QA			EL		
	64	128	256	64	128	256	64	128	256
FLAN-T5 + PTR	50.42	51.60	50.87	-	-	-	-	-	-
FLAN-T5 + Instruction tuning	-	-	-	92.09	94.23	93.71	89.26	95.17	98.06

Table 3
F1 scores for experiment 2, comparing the impact of number of instructions (e.g., K) across the three benchmark datasets (top row).

Train ↓ / Test →	EC+RC	QA	EL
EC+RC	**51.60**$^{\pm.47}$	78.36$^{\pm.86}$	57.40$^{\pm.16}$
QA	45.93$^{\pm.26}$	**94.23**$^{\pm.24}$	60.31$^{\pm.12}$
EL	45.04$^{\pm.26}$	68.22$^{\pm.70}$	95.17$^{\pm.39}$
EC+RC, QA	51.34$^{\pm.23}$	93.24$^{\pm.21}$	*58.96*$^{\pm.52}$
EC+RC, EL	51.21$^{\pm.45}$	*87.98*$^{\pm.31}$	**98.48**$^{\pm.29}$
QA, EL	*47.68*$^{\pm.23}$	93.69$^{\pm.24}$	97.61$^{\pm.14}$
all	50.55$^{\pm.70}$	94.10$^{\pm.27}$	98.19$^{\pm.32}$

Table 4
F1 scores of our previously best performing model: FLAN-T5 with 128-shot learning, on the different combinations of tasks we propose.

Finally, models that are tuned on all tasks do not outperform models tuned on two tasks in two out of three sets (only for QA does the full model perform better than models trained on two tasks).

6. Discussion

Our paper explored three different questions. First, *are "out-of-the-box" PLMs capable of generalizing learned behavior to labour market specific applications?* In order to answer this question, we created three self-supervised benchmarks from the ESCO taxonomy.

To answer this question, we performed zero-shot comparing between T5 and the instruction-tuned FLAN-T5, that has seen 1,836 additional tasks in prompts. Results showed that FLAN-T5 substantially outperforms T5 on two labour market-specific tasks, with a 49.7% increase in F1 score for QA, and 23.5% for EL. However in the EC+RC task where T5 outperforms FLAN-T5 by 3.53%. These findings confirm that overall, the instruction-tuned FLAN PLM benefits from having seen multiple tasks. The result for the EC+RC task can be explained by "misleading" patterns learned from the more general finetuning on named entity recognition (i.e., recognition of "Persons" and "Organizations", etc.). However, further investigations and ablation studies on general task tuning and its exact influence on the performance is needed for a more definite answer.

On the second question, whether *instruction and/or sub-prompt finetuning a PLM on a mixture of task-specific datasets could increase the performance on labour market specific benchmarks?*, we performed experiment 2, where we varied our K instruction samples for training our best-performing PLM: FLAN-T5. Results demonstrated that PTR-based finetuning with 128 examples leveraged the best performance. Overall, this yielded an 7.06% performance increase over the zero-shot performance of FLAN-T5. Additionally, further scaling of the number of examples, to 256, yielded only a 6.24% increase, suggesting no further performance increases for further scaling of the number of examples. Our results seem to indicate that using PTR with labour market specific examples yields improvements above and beyond the 1836 tasks FLAN-T5 was tuned on.

Lastly, we investigated *the effects of transfer learning across labour market specific tasks.*

Here, our results suggest that first, learning more tasks does yield increased performance on new, unseen tasks. At the same time, the best-performing models often were those that were trained on the evaluation task exclusively (for EC+RC and QA). Overall, unsurprisingly, directly learning the task at hand yields the best performing models, but the fact that multiple tasks improve performance for unseen tasks does suggest that the domain-specific knowledge that the PLMs receive in the tuning stage, do help solving the unseen task at hand. Prompt tuning on the QA and EL (i.e., instruction based finetuning) examples lead to a 3.14% improvement on the EC + RC task. Similarly, prompt tuning on the EC + RC and QA examples yielded a 1.58% increase in performance on the EL task, with an overall 4.54% increase over the zero-shot scenario. A possible explanation, "the ability to recognize whether an entity is an occupation or skill help discriminate whether two entities are not synonymous". However, training on all tasks did not seem to increase the overall performance on any of the tasks. We believe this is potentially caused by overlaps in learned behavior from these different labour market specific tasks and the 1836 tasks FLAN-T5 is already pre-trained on.

6.1. Implications

Finetuning PLMs is often an effective transfer mechanism in NLP. However, an entire new model is often required

for every task. Our results indicate that cost-efficient methods such as PTR and instruction-based finetuning can significantly increase the performance of PLMs on downstream labour market applications without introducing any additional model layers, manual annotations, and data augmentation.

Furthermore, our results suggest that while training on general tasks can increase the overall performance on labour market specific applications, providing the general models with labour market specific examples increases performance above and beyond the general finetuning.

6.2. Limitations

There are several limitations to the current study that should be considered. First, we only used one-to-one verbalizers between our classes and label words. Meaning that every class label is mapped to one respective label word. This would be a fruitful area for future research, e.g., occupation can also be rewritten as job, or work. Adding these alternatives to the label words will probably yield improved performance over the current one-to-one verbalizers.

Second, for the purpose of this initial exploration we focused primarily on binary classification tasks. As such, we did not incorporate the possibility for a non-existing relation in the PTR finetuning.

Third, while the underlying methods support multiple languages, we chose to conduct our experiments on English. In part because the descriptions used in the QA dataset are not complete for all 28 languages for which ESCO is available. A future study could assess the performance of PTR and instruction based finetuning without examples in other languages.

Lastly, this study primarily focused on the actual *isEssentialFor* and *isOptionalFor* relations as they were present in the ESCO taxonomy. As such, we did not implement the *reversed* and or negative relations, even though this was suggested to further increase performance.

7. Conclusion

In this study, we demonstrated that FLAN-T5 substantially outperforms T5 on the QA and EL tasks with respectively 49.7% and 23.5% F1 scores. However, on the remaining EC+RC task, T5 outperformed FLAN-T5 by 3.53%. Overall it seems that PLMs benefit from instruction based finetuning even on labour market specific benchmarks. However, if the task at hand is very different from the task at hand, it can potentially hurt performance, as demonstrated with the EC+RC tasks.

Furthermore, our results seem to indicate that using PTR with labour market specific examples yields improvements above and beyond the 1,836 tasks FLAN-T5 was

tuned on. Unsurprisingly, directly learning the task at hand leads to the best performing models. But, results also show prompt tuning on other labour market specific tasks can improve performance on unseen tasks. For example, prompt tuning on EC+RC and QA improved the performance on the EL task with 1.58%, and prompt tuning on QA and EL improved the performance on the EC+RC task by 3.14%.

There are several limitations to the current study; i) we solely used one-to-one verbalizer, ii) we focused primarily on binary classification tasks, iii) we only focused on English, and lastly we only used relations actually present in the ESCO taxonomy, meaning that we did not implement the reversed relations. Future studies could address the limitations of this study by using incrementing the amount of used label words, adding negative and reversed relations, and using ESCO to construct parallel datasets for all available languages.

References

[1] I. Khaouja, I. Kassou, M. Ghogho, A survey on skill identification from online job ads, IEEE Access 9 (2021) 118134–118153.

[2] M. Zhao, F. Javed, F. Jacob, M. McNair, Skill: A system for skill identification and normalization, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 29, 2015, pp. 4012–4017.

[3] E. M. Sibarani, S. Scerri, C. Morales, S. Auer, D. Collarana, Ontology-guided job market demand analysis: a cross-sectional study for the data science field, in: Proceedings of the 13th International Conference on Semantic Systems, 2017, pp. 25–32.

[4] J. Vrolijk, S. T. Mol, C. Weber, M. Tavakoli, G. Kismihók, M. Pelucchi, Ontojob: Automated ontology learning from labor market data, in: 2022 IEEE 16th International Conference on Semantic Computing (ICSC), IEEE, 2022, pp. 195–200.

[5] M. Zhang, K. N. Jensen, S. D. Sonniks, B. Plank, Skillspan: Hard and soft skill extraction from english job postings, arXiv preprint arXiv:2204.12811 (2022).

[6] M. Zhang, R. van der Goot, B. Plank, Escoxlm-r: Multilingual taxonomy-driven pre-training for the job market domain, arXiv preprint arXiv:2305.12092 (2023).

[7] M. de Groot, J. Schutte, D. Graus, Job posting-enriched knowledge graph for skills-based matching, in: RecSys in HR 2021, Amsterdam, Netherlands, 2021.

[8] U. P. K. Kethavarapu, S. Saraswathi, Concept based dynamic ontology creation for job recommendation system, Procedia Computer Science 85 (2016) 915–921. URL: https://www.sciencedirect.com/science/

article/pii/S1877050916306329. doi:`https://doi.org/10.1016/j.procs.2016.05.282`, international Conference on Computational Modelling and Security (CMS 2016).

[9] A. Weichselbraun, R. Waldvogel, A. Fraefel, A. van Schie, P. Kuntschik, Building knowledge graphs and recommender systems for suggesting reskilling and upskilling options from the web, Information 13 (2022). URL: https://www.mdpi.com/2078-2489/13/11/510. doi:`10.3390/info13110510`.

[10] X. Han, W. Zhao, N. Ding, Z. Liu, M. Sun, Ptr: Prompt tuning with rules for text classification, AI Open 3 (2022) 182–192.

[11] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma, et al., Scaling instruction-finetuned language models, arXiv preprint arXiv:2210.11416 (2022).

[12] T. Gao, A. Fisch, D. Chen, Making pre-trained language models better few-shot learners, arXiv preprint arXiv:2012.15723 (2020).

[13] T. Schick, H. Schütze, Exploiting cloze questions for few shot text classification and natural language inference, arXiv preprint arXiv:2001.07676 (2020).

[14] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training (2018).

[15] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:`10.18653/v1/N19-1423`.

[16] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized BERT pretraining approach, CoRR abs/1907.11692 (2019). URL: http://arxiv.org/abs/1907.11692. `arXiv:1907.11692`.

[17] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (2020).

[18] M. E. Peters, M. Neumann, R. L. Logan IV, R. Schwartz, V. Joshi, S. Singh, N. A. Smith, Knowledge enhanced contextual word representations, arXiv preprint arXiv:1909.04164 (2019).

[19] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, Advances in neural information processing systems 26 (2013).

[20] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, Q. Liu, Ernie: Enhanced language representation with in-formative entities, arXiv preprint arXiv:1905.07129 (2019).

[21] M. Yasunaga, A. Bosselut, H. Ren, X. Zhang, C. D. Manning, P. S. Liang, J. Leskovec, Deep bidirectional language-knowledge graph pretraining, Advances in Neural Information Processing Systems 35 (2022) 37309–37323.

[22] J. Liu, Z. Zhang, Z. Guo, L. Jin, X. Li, K. Wei, X. Sun, Kept: Knowledge enhanced prompt tuning for event causality identification, Knowledge-Based Systems 259 (2023) 110064.

[23] X. Chen, N. Zhang, X. Xie, S. Deng, Y. Yao, C. Tan, F. Huang, L. Si, H. Chen, Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction, in: Proceedings of the ACM Web conference 2022, 2022, pp. 2778–2788.

[24] L. Cui, Y. Wu, J. Liu, S. Yang, Y. Zhang, Template-based named entity recognition using BART, CoRR abs/2106.01760 (2021). URL: https://arxiv.org/abs/2106.01760. `arXiv:2106.01760`.

[25] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, Q. V. Le, Finetuned language models are zero-shot learners, arXiv preprint arXiv:2109.01652 (2021).

[26] S. Roller, D. Kiela, M. Nickel, Hearst patterns revisited: Automatic hypernym detection from large text corpora, arXiv preprint arXiv:1806.03191 (2018).

[27] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al., Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations, 2020, pp. 38–45.

[28] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035.

[29] N. Ding, S. Hu, W. Zhao, Y. Chen, Z. Liu, H.-T. Zheng, M. Sun, Openprompt: An open-source framework for prompt-learning, arXiv preprint arXiv:2111.01998 (2021).

[30] N. F. Liu, A. Kumar, P. Liang, R. Jia, Are sample-efficient nlp models more robust?, arXiv preprint arXiv:2210.06456 (2022).