Universität Zürich^UZH

Master Thesis

for obtaining the academic degree

Master of Informatics

from the Faculty of Business, Economics, and Informatics

Adapting Pre-trained Transformer Language Models for Mapping Texts on Domain-Specific Ontologies

Author: Tanmay Chimurkar

Matriculation-Nr: 20-745-907

Referentin/Referent: Prof. Dr. Martin Volk

Supervisor: Dr. Simon Clematide

Institut für Informatik

Abgabedatum: (26.03.2023)

Abstract

This master thesis explores domain adaptation methods for pre-trained Large Language Models (LLMs) to map natural language mentions from a text genre onto a target domain ontology based on cosine similarity in a semantic vector space. For the thesis, the input mentions are skill requirement mentions extracted from Swiss job ad postings written in German or English, and the target domain onto which these terms have to be mapped is the European Skills, Competences, Qualifications and Occupations (ESCO) ontology. The objective of this task is to track changes in the labor market and help recruiters fill positions based on skill requirements fulfilled by candidates. The thesis explores three methods: Masked Language Modelling, Multiple Negative Ranking Loss, and binary classification method for further pre-training in order to adapt LLMs to a target domain ontology. Experiments were conducted on 15 model variants using different input data and starting models. Two gold standard datasets, one consisting of randomly selected skill requirement mentions, and the other specifically crafted from challenging cases, were used for evaluating model performance. The evaluations were created by annotating the top suggestions made by our model variants. Mean Average Precision (MAP) scores were computed based on human annotations of the suggestions, made by each model variant for each term in the gold standard datasets. MAP is used as an evaluation metric since more than one mapping might be correct or acceptable, and a good ranking of the appropriate ontology concepts can be measured via this metric. The MNR models with the hard negative sampling strategy, wherein the negative samples are taken with lexical and semantic similarities to the anchor term, and domain adaptation on both the job-ads data and the ESCO ontology data were found to be the best-performing model variants for both the English and German languages. The thesis concludes that domain adaptation on both the input texts and the target domain is beneficial for mapping mentions from the input genre onto the target domain. It also suggests that using a hard negative sampling method for creating the MNR data is beneficial compared to a random negative sampling method.

Zusammenfassung

Diese Masterarbeit untersucht Methoden zur Domänenanpassung für vortrainierte Large Language Models (LLMs), um natürlichsprachliche Erwähnungen aus einem Textgenre auf eine Zieldomänenontologie abzubilden, basierend auf Kosinusähnlichkeit in einem semantischen Vektorraum. In dieser Arbeit sind die Input-Erwähnungen Erwähnungen von Qualifikationsanforderungen, die aus Schweizer Stellenanzeigen in deutscher oder englischer Sprache extrahiert wurden, und die Zieldomäne, auf die diese Begriffe abgebildet werden müssen, ist die European Skills, Competences, Qualifications and Occupations (ESCO) Ontologie. Ziel dieser Aufgabe ist es, Veränderungen auf dem Arbeitsmarkt zu verfolgen und Personalverantwortliche bei der Besetzung von Stellen auf der Grundlage der von den Bewerbern erfüllten Qualifikationsanforderungen zu unterstützen. In dieser Arbeit werden drei Methoden untersucht: Masked Language Modelling, Multiple Negative Ranking Loss und eine binäre Klassifizierungsmethode für weiteres Pre-Training, um LLMs an eine Zieldomänen-Ontologie anzupassen. Es wurden Experimente mit 15 Modellvarianten unter Verwendung unterschiedlicher Eingabedaten und Ausgangsmodelle durchgeführt. Zur Bewertung der Modellleistung wurden zwei Goldstandard-Datensätze verwendet, von denen einer aus zufällig ausgewählten Erwähnungen von Qualifikationsanforderungen besteht und der andere speziell aus anspruchsvollen Fällen zusammengestellt wurde. Die Auswertungen wurden durch Annotation der besten Vorschläge unserer Modellvarianten erstellt. Die mittlere durchschnittliche Präzision (MAP) wurde auf der Grundlage der menschlichen Annotationen der Vorschläge berechnet, die von jeder Modellvariante für jeden Begriff in den Goldstandard-Datensätzen gemacht wurden. MAP wird als Bewertungsmaßstab verwendet, da mehr als eine Zuordnung richtig oder akzeptabel sein kann und ein gutes Ranking der entsprechenden Ontologiekonzepte anhand dieses Maßstabs gemessen werden kann. Die MNR-Modelle mit der Hard-Negative-Sampling-Strategie, bei der negative Stichproben mit lexikalischen und semantischen Ähnlichkeiten zum Ankerterminus genommen werden, und die Domänenanpassung sowohl für die Daten der Stellenanzeigen als auch für die Daten der ESCO-Ontologie erwiesen sich sowohl für die englische als auch für die deutsche Sprache als die leistungsstärksten Modellvarianten. Die Arbeit kommt zu dem Schluss, dass die Domänenanpassung sowohl für die Eingabetexte als auch für die Zieldomäne von Vorteil ist, um Erwähnungen aus dem Eingabegenre auf die Zieldomäne abzubilden. Sie legt auch nahe, dass die Verwendung einer harten negativen Stichprobenmethode für die Erstellung der MNR-Daten im Vergleich zu einer zufälligen negativen Stichprobenmethode von Vorteil ist.

Acknowledgement

I am deeply grateful to Dr. Simon Clematide and Ann-Sophie Gnehm for their invaluable help and guidance throughout my Master's thesis. Dr. Clematide provided exceptional support, structuring the report in a way that made it more accessible to readers and proofreading it meticulously. Ann-Sophie Gnehm's input, ideas, and insights have been instrumental in refining the research output, and I appreciate her assistance throughout the research process.

I am also grateful to Kartikey Sharma, who proofread the report and provided valuable suggestions and comments, as well as to Dr. Clematide for his helpful feedback and suggestions.

Furthermore, I want to express my appreciation to the University of Zurich for providing me with an excellent academic experience during my Master's studies. The university's distinguished faculty, world-class facilities, and supportive environment have been invaluable to my development as a researcher.

Lastly, I would like to extend my heartfelt thanks to my parents, my brother, and my fiancé for their unwavering support, encouragement, and understanding throughout my thesis. Their love, care, and encouragement have been an essential source of motivation, enabling me to achieve my goals.

Contents

List of Figures

List of Tables

List of Acronyms

ESCO	European Skills, Competences, Qualifications and Occupations
KG	Knowledge Graph
LLM	Large Language Models
MLM	Masked Language Modelling
MNR	Multiple Negative Ranking Loss
NER	Named Entity Recognition
NLP	Natural Language Processing
RDF	Resource Description Framework
URI	Uniform Resource Identifier

1 Introduction

With the advancements in technology in the field of NLP, many tasks related to languages ranging from sentiment analysis and question answering to text classification and machine translation are made easier. One of the main advancements that made these tasks easier was the introduction of the transformer models, which introduced the attention mechanism (Vaswani et al. [2017]). With the introduction of the attention mechanism, many other transformer models were created, with BERT (Devlin et al. [2019]) being the first transformer model. However, language itself is complex to understand, and for specific tasks like machine translation or text summarization, the language model used needs to have a good understanding of how language is structured. And if we decide to train a transformer for every specific task, then instead of understanding language the model would try to understand the task-specific pattern for getting the best response for a task. An alternative to this approach is to instead pre-train a Large Language Model (LLM) using a very general task such as predicting masked words or the next word. The general language understanding gained from these tasks can then be transferred (transfer learning[1]) to more specific tasks where less training data is available.

Further pre-training and fine-tuning large pre-trained LLMs to a particular domain or task has proved beneficial as part of domain adaptation ([Gururangan et al., 2020, 8342–8360]). However, there are numerous methods available to fine-tune LLMs for a particular domain or task, and using the methods to adapt an LLM for a particular domain depends on the formulation of the available domain data and what tasks we want to achieve from that domain data.

The main motivation of this thesis is to explore which methods prove to be beneficial for adapting a pre-trained LLM for a particular domain and tasks. The task that this thesis focuses on is the task of classifying text mentions from an input domain onto a target domain, where the target domain is an ontology from a particular domain. The input and the target domains need not be the same, but they have to be relatable to each other on some common ground. An example of the type of

[1] Transfer Learning: `https://en.wikipedia.org/wiki/Transfer_learning`

problem where text was mapped onto an ontology includes the SGD policy mapper project by the European Union[2], wherein this tool counts the EU policies linked to each goal and target while specifying their connections, enabling the identification of crosscutting initiatives - policies that address multiple goals. Another such example is in the Paper Mirhosseini et al. [2014], wherein NLP methods and IR[3] methods are used for mapping a particular concept onto the SNOMED CT[4] medical domain ontology. Other areas where mapping text onto an ontology can be beneficial are banking and finance systems, customer support, etc. Thus we can see that the task of mapping text mentions from an input domain onto a target ontology can be beneficial and applicable in various application scenarios.

For this thesis, the main task is to classify worker skill requirements that are extracted from job ads data. The job ads data is from the German-speaking job market in Switzerland, with job ads containing expressions specific to Switzerland in them. Such a mapping would enable recruiters and job seekers to analyze the current skill requirements in the job market and be able to fulfill those requirements easily by mapping candidate applications onto the required skills based on the current job market requirements. The mapping of job ad terms has to be done in an unsupervised approach, with the available classes to map onto the skills and competencies as defined in the ESCO[5] (European Skills, Competences, Qualifications, and Occupations) ontology, which acts as a collection of the skills and competences terms that are usually seen in candidate job applications. The unsupervised mapping is to be done using the semantic similarities, rather than lexical similarity, between the terms in job ads data and the terms available in the ESCO ontology. Figure 1 represents a flow chart of how the job ad data and the ESCO ontology data are combined via pre-training and fine-tuning of LLMs, and then those LLMs are used for the end task of mapping terms onto the ESCO ontology, and Table 1 gives an example of the actual job ad mentions and the matching ESCO concepts from the ontology. Since the task of mapping job ad terms onto ESCO terms is very dependent on the domain data, it is thus beneficial to adapt LLM to the domain of ESCO rather than using an out-of-the-box LLM for mapping.

[2] SGD Policy mapper: https://knowsdgs.jrc.ec.europa.eu/sdgmapper

[3] Information Retrieval: https://en.wikipedia.org/wiki/Information_retrieval

[4] SNOMED CT Ontology: https://bioportal.bioontology.org/ontologies/SNOMEDCT

[5] ESCO: https://esco.ec.europa.eu/en/about-esco/what-esco

Job ad mentions	Class numeric	Class label
** *Second Level Support* ** Koordination, Unterstützung von Key Usern	skill/S5.2.3	resolving computer problems
** *EDV* ** DOS, Windows	skill/T2.3	Handhaben von Problemen
FAGE ** *FA SRK* **	isced-f/0913	Krankenpflege und Hebammenausbildung

Table 1: Example of mapping of job ad mentions with concepts from the ESCO ontology. The terms in italics are the central job ad mentions, and the other terms are surrounding context terms passed along with the job ad mentions. The class numeric is the tag of the concept class to which the skill belongs to, and the class label shows the label for that particular class.

1.1 Research Questions

Since the motivation of this thesis is to find methods that work well for adapting a large LLM to a particular domain, the research questions that shall be answered in this thesis, are:

1. Which pre-training and fine-tuning methods such as Masked Language Modelling (MLM), Multiple Negative Ranking (MNR), and classification is helpful for domain adaptation and the end-task of mapping terms onto an ontology?

2. Does hard negative sampling, wherein a negative sentence pair is lexically and semantically similar to the anchor term, yield better results than using random negative sampling for domain adaptation using the Multiple Negative Ranking loss?

3. Does a sentence classification approach using labels derived from the structure of an ontology prove to be beneficial for the domain adaptation of an LLM?

4. Would using LLMs that have been further pre-trained on the input texts, be necessary or improve the target task of mapping terms onto an ontology?

The goals to answer the above-defined research questions need data from both the job ads and the ESCO ontology. In this regard, the first goal is to extract the linguistic material (terms, descriptions) and conceptual structure from the ESCO ontology and the job ads data so that it can then be restructured as per the task that is used for fine-tuning an LLM for domain adaptation. The next goal is to understand the relations within the ESCO data since the data has a multi-hierarchy

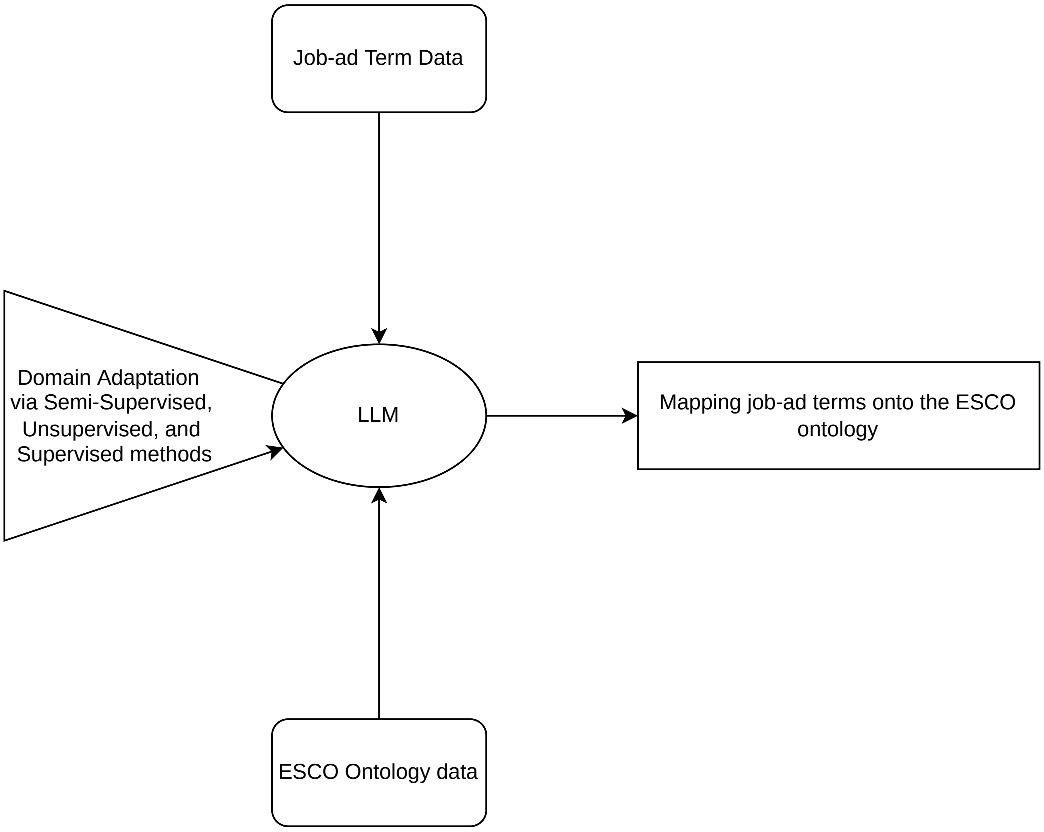

Job-ad Term Data
Domain Adaptation via Semi-Supervised, Unsupervised, and Supervised methods
LLM
Mapping job-ad terms onto the ESCO ontology
ESCO Ontology data

Figure 1: Flow of mapping job ad terms onto ESCO Ontology

structure. Once the data is available from the ESCO ontology, we can then perform pre-training and fine-tuning tasks[6] and evaluate the results of the model to check if the methods used are beneficial for the end task of mapping terms onto the ontology using an out-of-the-box LLM as a baseline. Ablation studies to check the effect of different methods will also be conducted during the evaluation to answer which methods, in sequence or in combination with other methods, are the most beneficial for domain-adapting pre-trained transformer models.

1.2 Thesis Structure

The thesis is subdivided into 7 parts, wherein this part gives a brief introduction of the motivation of the thesis and the research questions that it tries to answer.
Chapter 2 Introduces the related work for similar domains and tasks. We will look at problems of a similar type as the one in this thesis and the methods used to solve

[6] Code link: `https://github.com/tanmaychimurkar/ma-thesis`

those problems.

Chapter 3 Explains the domain data that is used for fine-tuning an LLM. It will be explained in greater detail what the ESCO ontology looks like and what relations it consists of. This section will also cover methods used for extracting the data via a pipeline that has been designed as part of this thesis to gather data from the ontology for a particular language.

Chapter 4 Explains the methods used for fine-tuning LLMs for domain adaptation on the ESCO ontology and the job ads data.

Chapter 5 Gives a brief overview of the training procedure followed for building the pipelines necessary for fine-tuning LLMs.

Chapter 6 Gives a detailed explanation of the results and their implications.

Chapter 7 Explains the conclusions from the experiments on the data and highlights future work.

2 Related Work

Specifically for the task of classifying job-ad terms onto the ESCO ontology, the recent work done by Zhang et al. [2022] explores methods for fine-tuning transformer LM on ESCO data. They use the *Kompetencer* dataset, which contains job ads in Danish and English language annotated at the span-level of skills and competencies. They explore zero-shot and few-shot cross-lingual transfer learning on the out-of-the-box English BERTbase model and a BERTbase model that is continuously pre-trained on English job-posting data. They do a similar transfer learning task on an out-of-the-box Danish BERT model and the same model is again fine-tuned on the job posting data. An overview of the methodology they follow in is shown in Figure 2.

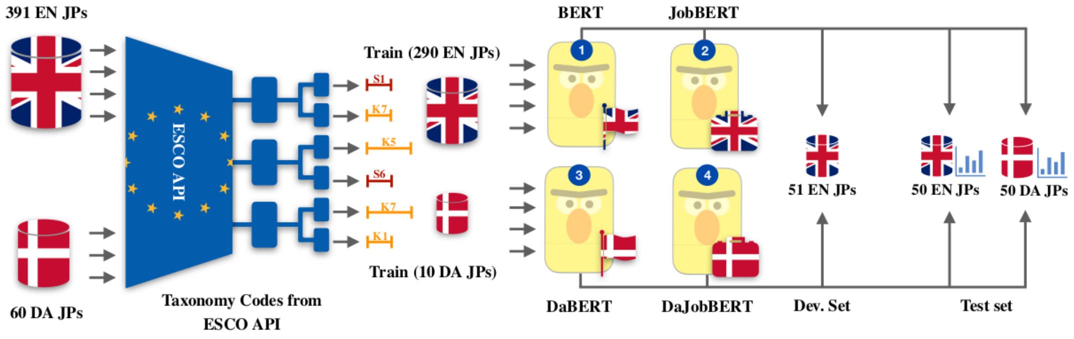


Figure 2: The pipeline followed in Zhang et al. [2022]. Job postings in English and Danish language are selected, and they label them with the respective labels from the ESCO ontology data. After this step, LLMs for English and Danish language are trained separately using zero-shot and few-shot learning methods, and their performances are checked via dev sets of job postings in both languages.

They mention that domain-adaptive pre-training provides short-term gains with little cost, but there needs to be a lot of unlabelled data from the domain. On the other hand, if the data has to be labeled, then the costs go up.

For domain adaptation with a small amount of available data, Tunstall et al. [2022] explores methods for efficient few shot-learning without prompts. The work done in this paper proposes SETFIT (Sentence Transformer Fine-tuning), wherein a pre-trained sentence transformer is first fine-tuned on a small amount of training data and is then used to generate rich word embeddings. They used no prompts while fine-tuning the sentence transformer models, and used few-shot learning methods with a limited amount of data for fine-tuning. An overview of the training methodology is shown in Figure 3. They use triplet data to fine-tune the sentence transformer, with the data being in the form of (x_i, y_i, label) format, where x_i and y_i are sentence pairs with the label indicating if the sentences are related to each other. The embeddings for the sentence pairs are generated and combined, and the classification head uses these embeddings for training based on the labels provided.

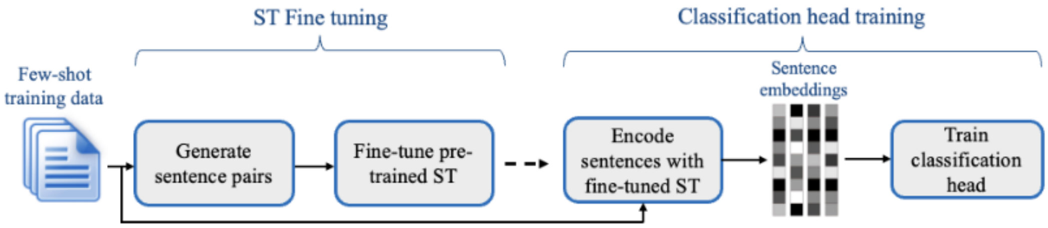


Figure 3: The pipeline followed in Tunstall et al. [2022]. A sentence transformer is initially fine-tuned, and a classification head is trained using the encoded data obtained from the fine-tuning step

The work done in Li et al. [2022] mentions multi-task pre-training of LLMs via a modified version of the MLM task on an ontology. In this paper, they used the Masked Entity Modelling (MEM) task wherein only the entity sequence of the input is masked instead of randomly masking words from the input. They further use Masked Relation Modelling (MRM) wherein they mask only the relation words between two entities. They use these two novel methods alongside the MLM method to fine-tune LLMs on a particular ontology for domain adaptation and link prediction in the ontology. The methodology used in is in Figure 4.

In the work done in Kim et al. [2020], the authors suggest combining the tasks of relationship prediction and relevance ranking tasks with target link prediction to improve an LLM's ability to understand the domain of the ontology and perform correctly even if there are lexical similarities between the input term and terms in the ontology. They propose KG-BERT, a language model fine-tuned via the link prediction and the relationship prediction tasks on the data of an ontology. The proposed methodology is shown in Figure 5.

The work done in Li et al. [2019] reflects on methods for sampling negative data. They propose 4 different model adaptive negative sampling methods, based on dif-

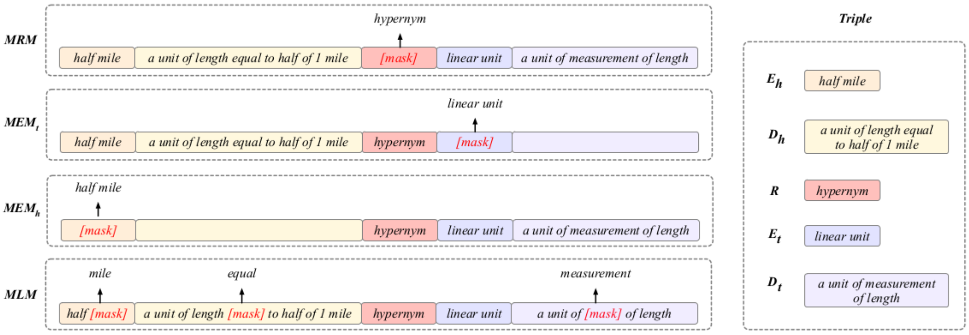

MRM
hypernym
half mile
a unit of length equal to half of 1 mile
[mask]
linear unit
a unit of measurement of length
MEM_t
linear unit
half mile
a unit of length equal to half of 1 mile
hypernym
[mask]
MEM_h
half mile
[mask]
hypernym
linear unit
a unit of measurement of length
MLM
mile
equal
measurement
half [mask]
a unit of length [mask] to half of 1 mile
hypernym
linear unit
a unit of [mask] of length
Triple
E_h
half mile
D_h
a unit of length equal to half of 1 mile
R
hypernym
E_t
linear unit
D_t
a unit of measurement of length

Figure 4: The MEM, MRM, and MLM methods used for domain adaptation. Image Source Li et al. [2022]

ferent thresholds used to determine if a sample is a negative sample or not. They also use matching models which are used to compare a possible negative term to the input term to determine if they are semantically similar rather than being only lexically similar. The methods explored in this paper are extendible to the MNR loss strategy for mining negative samples and creating a triplet pair dataset.

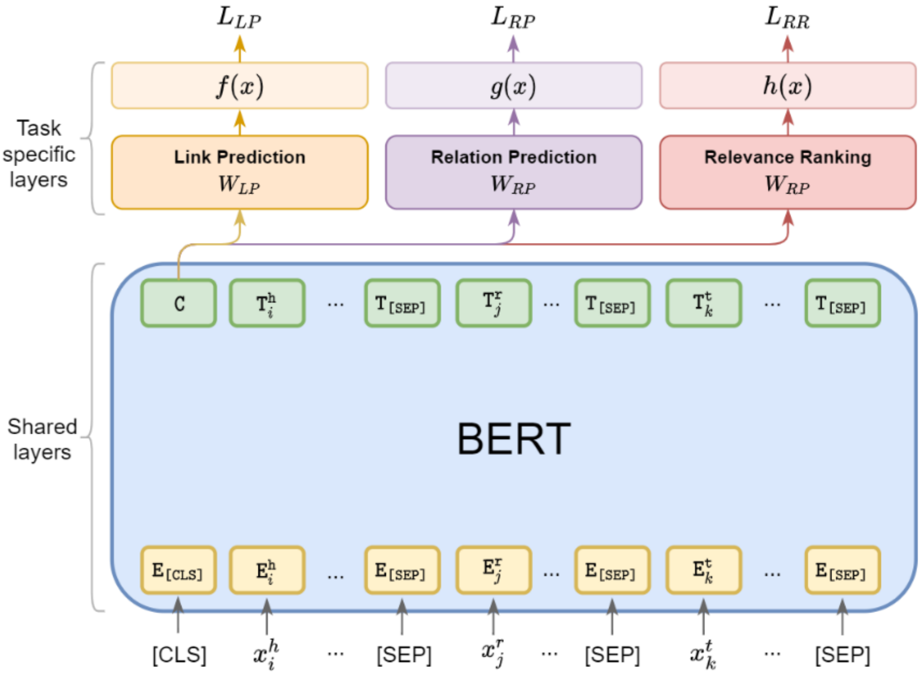


Figure 5: Multi-task learning methods for Knowledge Graph completions and domain-adaptation

3 Data

Structured information models allow us to represent and organize knowledge in a structured and organized manner. These models often define knowledge with a typical set of rules and relationships. These models can be used to represent all types of data, ranging from physical objects to abstract concepts and events. Examples of common structured information models are:

1. **ER Models**: In typical databases, information is stored in the form of entity-relationship models, which is a highly-structured data model. It is typically used in database design, wherein relationships between different entities are modeled using an ER diagram.

2. **Resource Description Framework (RDF)**: This is the typical structured model used to describe information on the web. This framework forms the basics of the Semantic web, and data in this framework is typically represented in triplets consisting of subject, predicate, and object.

3. **Ontologies**: These are models that define concepts and the relationships between different entities. Ontologies are usually defined for a domain to represent knowledge about that particular domain.

4. **Knowledge Graphs**: These are graph-based data models that are used to store and represent information as nodes and relationships between them. The structure of a knowledge graph helps define a rich semantic relationship between node pairs, and also allows for an intelligent search of nodes based on their relationships.

Structured information models are used in major applications of information storage and retrieval. Let us now look at knowledge graphs in more detail as our data structure comprises a knowledge graph.

3.1 Knowledge Graph

As per the Resource Description Framework (RDF)[1] specification, all the data in Knowledge Graphs (KG)[2] is represented in the form of *(entity, relation, entity)* triplets. This triplet data can be interpreted in plain language as follows: the first *entity* is associated with the second *entity* with a relationship *relation*. In RDF terminology, all entities have a Uniform Resource Identifier (URI)[3], and each URI is unique for each individual entity in a KG.

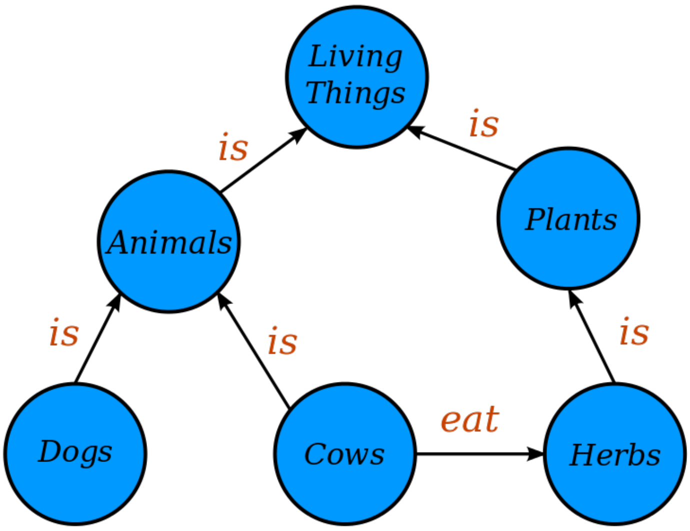


Figure 6: Structure of a Knowledge Graph. Each blue-colored node represents an *entity*, and each directed arrow represents a *relationship* between two entities. Image Source: `https://en.wikipedia.org/wiki/Knowledge_graph`

Figure 6 shows the basic structure of a KG. From this basic structure, we can see that each node has a relationship to another node, where the relationship is defined by the arrow line. For any node, we can thus interpret the relationship it has with other nodes. For example, for the node *Cows*, we can interpret the following: The entity Cows '*is*' an entity *Animals*, and also '*eats*' *Herbs*.

[1] Resource Description Framework: `https://www.w3.org/RDF/`

[2] Knowledge Graph: `https://en.wikipedia.org/wiki/Knowledge_graph`

[3] Uniform Resource Identifier: `https://en.wikipedia.org/wiki/Uniform_Resource_Identifier`

3.2 ESCO Ontology

Our main source of data is the ESCO[4] ontology, which is available as a KG. ESCO is a European multilingual classification of Skills, Competences, and Occupations, wherein the classifications are available in 28 languages. ESCO can be seen as a multilingual terminology to look up a particular set of skills, competencies, or occupations from the European Union labor market. Each particular skill or occupation in the ESCO ontology is a *'concept'*, and each concept has properties and relations defined for it that are uniform for easy understanding, parsing, and usage by electronic systems. Moreover, the ESCO ontology also has a multi-hierarchical structure, meaning that one concept can belong to more than one *parent class* as shown in Figure 6. This information is very crucial for us since by using the multi-hierarchy structure, we can classify the concepts from ontology into multiple *class* level(s) that it belongs to. Furthermore, different classes can be suggested for a particular concept based on the context it falls under.

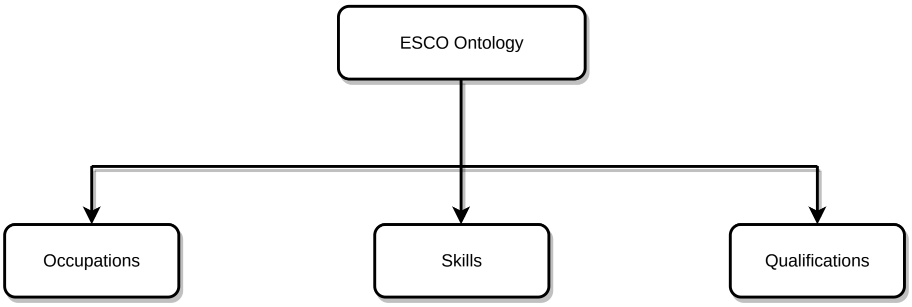


Figure 7: The classification of the whole ESCO ontology into the 3 major pillars

In Figure 7, the first major levels of classification of the ESCO ontology are highlighted. Based on this figure, we can see that the ESCO ontology is subdivided into the 3 main *pillars*: Occupation, Skills, and Qualifications. For our case, we want to map the job-ad terms to the closest matching concept from the ESCO ontology, and for this purpose, we focus primarily on the *Skill* pillar from the ESCO ontology. This is because the terms that we are interested in from the job-ads data are the terms that most closely classify as skills in the ESCO ontology. Inside the ESCO Skill pillar, we have the following 4 main categories as shown in Figure 8 related to skills: knowledge skills, transversal skills, language skills, and skills. These 4 categories together make up the ESCO Skill pillar. Each of these 4 categories have many classes that they are divided into, and their categorizations are shown in Figure 9 to 12. The 4 subsections below give a description of each of the 4 categories that

[4] ESCO: `https://esco.ec.europa.eu/en`

fall under the Skill pillar.

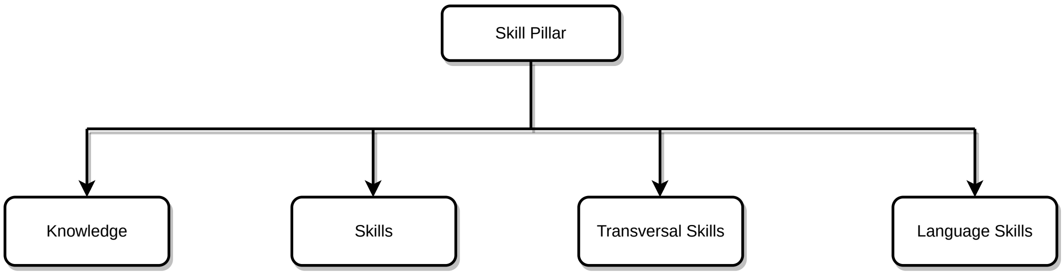


Figure 8: The classification of the Skill pillar into 4 skill categories

3.2.1 Knowledge Class

The *Knowledge* class contains concepts that are related to the knowledge that is needed in a particular field and contains the highest abstract term that defines a particular field. The examples of types of concepts that are contained in the knowledge class are shown in Table 2.

Concept Term	**Concept Class**
Algebra	Natural Sciences, Mathematics, and Statistics
Military Aviation	Services

Table 2: Examples of Knowledge Class from the ESCO ontology

The 12 top-level concepts that the knowledge class consists of are shown in Figure 9.

3.2.2 Skill Class

The *Skills* class contains the actual skills that are required to do a task for a concept from the knowledge class. Examples of concepts that fall into the skills class are in Table 3.

The 8 top-level concepts that the skill class consists of are shown in Figure 10.

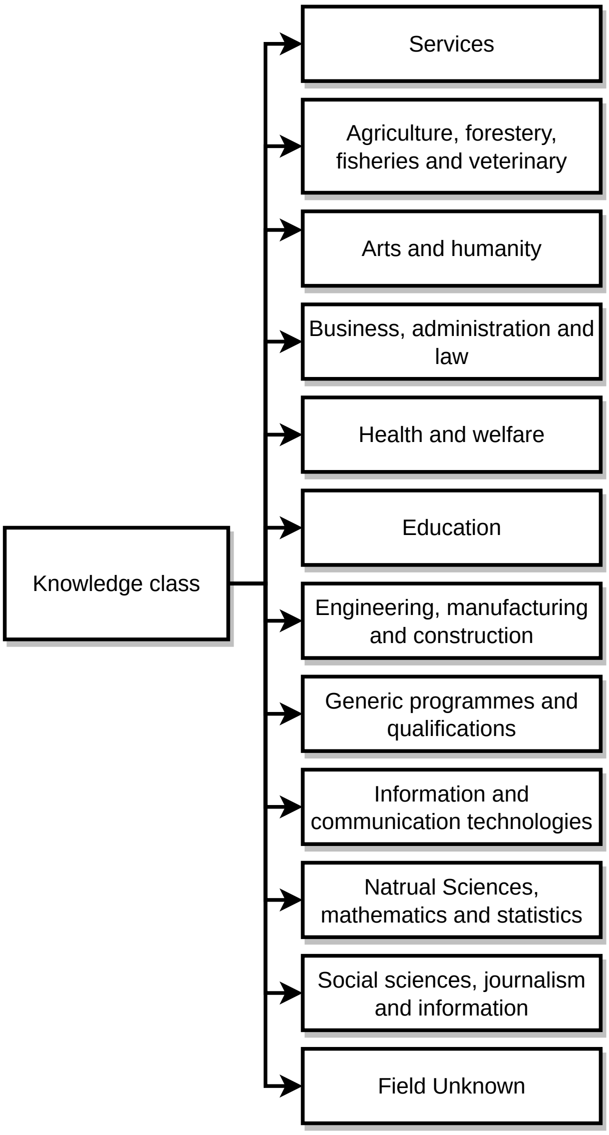

Knowledge class
Services
Agriculture, forestery, fisheries and veterinary
Arts and humanity
Business, administration and law
Health and welfare
Education
Engineering, manufacturing and construction
Generic programmes and qualifications
Information and communication technologies
Natrual Sciences, mathematics and statistics
Social sciences, journalism and information
Field Unknown

Figure 9: The highest level of concepts that are contained in the *Knowledge* class

3.2.3 Transversal Skill Class

The *Transversal skills* class contains concepts that act as competence to do a particular task. Examples of concepts that fall into the transversal skills are shown in Table 4.

The 6 top-level concepts that the skill class consists of are shown in Figure 11.

Concept Term	Concept Class
Manage Food Manufacturing Laboratory	Management Skills
Maintain Barrels	Constructing

Table 3: Examples of Skill Class from the ESCO ontology

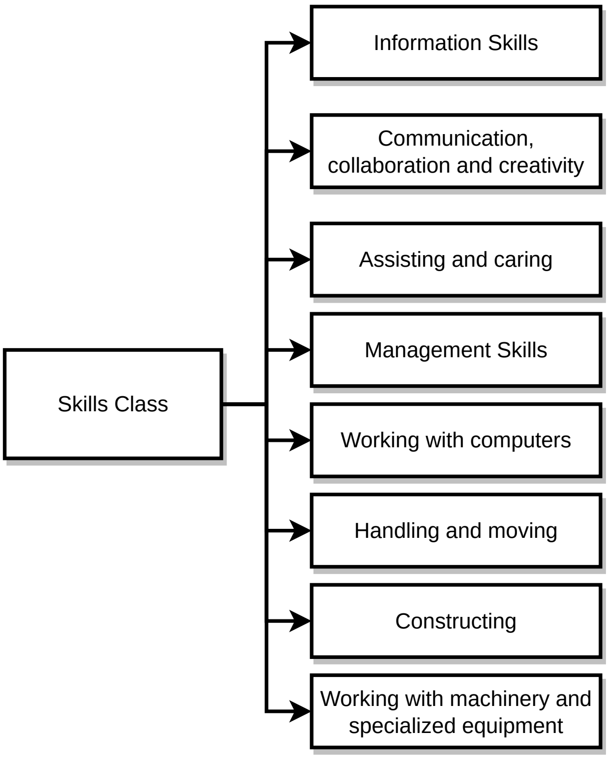


Figure 10: The highest level of concepts that are contained in the *Skills* class

3.2.4 Language Class

The *Language* class contains skills that are very specific to the usage of languages in handling tasks. Examples of concepts that fall into this class are shown in Table 5

The 2 top-level concepts that the skill class consists of are shown in Figure 11.

3.2.5 ESCO Ontology Vocabulary

In the RDF data model, data is often represented in a turtle file. A turtle file is a format to represent data in the form of a directed graph as a set of nodes, which

Concept Term	Concept Class
Calculate Probabilities	Core Skills and Competences
Solve Problems	Thinking Skills and Comptences

Table 4: Examples of Transversal Class from the ESCO ontology

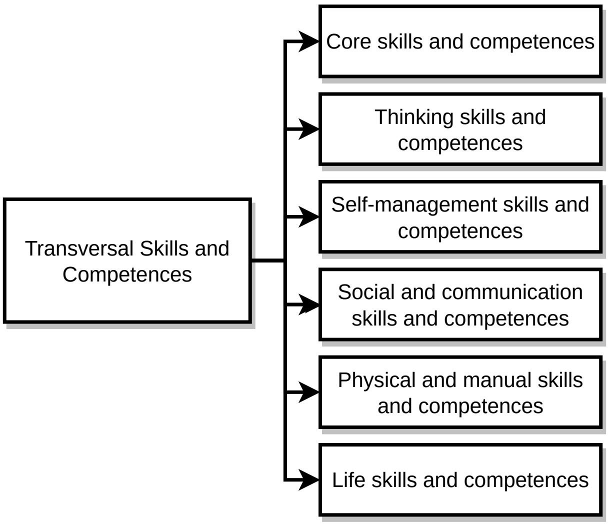

Transversal Skills and Competences
Core skills and competences
Thinking skills and competences
Self-management skills and competences
Social and communication skills and competences
Physical and manual skills and competences
Life skills and competences

Figure 11: The highest level of concepts that are contained in the *Transversal Skills and Competences* class

are vertices, which are connected by edges, forming links between nodes. In a turtle file, each relation represented by the nodes and the edges consists of triplets of the form subject, predicate, and object. The subject and the object are usually nodes that represent URIs of entities of the data that the turtle file represents, and the predicate is the edge representing the relationship between nodes. The relationships that are described by the predicates in a turtle file are referred to as vocabularies, and these vocabularies differ for a turtle file based on the data it is representing.

As the ESCO ontology data is available as an RDF turtle file[5], each entity inside the ontology needs to follow a vocabulary that gives every entity a relationship that can be associated with it. For this purpose, the turtle file of the ESCO ontology defines its own vocabulary based on the official ESCO data model[6] of ESCO ontology.

Alongside the ESCO vocabulary, the turtle file also contains many vocabularies as

[5] RDF turtle format: `https://www.w3.org/TR/turtle/`

[6] ESCO data model: `https://ec.europa.eu/esco/lod/static/model.html`

Concept Term	Concept Class
Understand Spoken English	English
Write German	German

Table 5: Examples of Language Class from the ESCO ontology

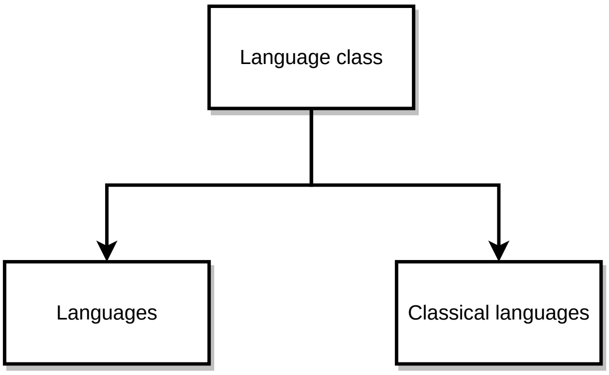


Figure 12: The highest level of concepts that are contained in the *Language* class

per the W3C specification[7], like the skos[8], the XLMSchema[9], vocabularies as per the DublinCore specification like dct[10], iso-thes[11]. All these externally established vocabularies, along with ESCO's vocabulary, are used to enhance the structure of the ESCO turtle structure so that basic CRUD[12] operations on the ESCO turtle file are relatively easy to make.

3.2.6 ESCO Data Model

Since the ESCO ontology data is available as an RDF turtle file, we created a simplified visual version of the structure of the ontology that takes into account all the pillars of the ESCO ontology. From this visual structure, we can better understand how each *'concept'* inside the ontology is related to other entities, and what relationships and properties each entity has.

[7] World Wide Web Consortium: `https://www.w3.org/`

[8] Simple Knowledge Organization System: `https://www.w3.org/2009/08/skos-reference/skos.html`

[9] XML Schema: `https://www.w3.org/2001/XMLSchema#`

[10] DCMI Metadata Terms: `https://www.dublincore.org/specifications/dublin-core/dcmi-terms/`

[11] skos-thes: `http://pub.tenforce.com/schemas/iso25964/skos-thes`

[12] CRUD operations: `https://en.wikipedia.org/wiki/Create,_read,_update_and_delete`

As per the vocabulary of the ESCO data model itself, we can break it down into a simplification of the original data model as shown in Figure 13. For our case, we are using ESCO V1.1 as the main source of data, and the data model associated with this version. As per the figure, for our case, the central most entity is of type *‘Memberconcept’*, which we will refer to as a ‘concept’. A concept is of type *‘Skill‘* or *‘Occupation’* as per the ESCO vocabulary, and of type *‘concept’* as per the skos vocabulary. As per the official ESCO data model, the ‘Memberconcept‘ entities are all the entities that inherit from the ‘Skill’ and the ‘Occupation’ pillars of the ESCO structure. There are a total of 14,529 concepts of type ‘Skill’ and 3008 concepts of type ‘Occupation’ in the ESCO ontology. This number of concepts is the same for the English and the German languages, as these concepts each have a unique URI, and only have properties associated with it in each language. Each concept could have the following descriptive properties related to it as per the skos vocabulary: prefLabel, altLabel, hiddenLabel, and description, where each of the properties respectively describes the preferred label of the concept, its alternate label, its hidden label and the description used to describe that concept. An example of the properties associated with each concept is shown in Table 6.

Concept Property	**Property Value**
concept URI	skill/059b2748-e1df-43d2-8a08-0331fb8d0ddf
Preferred Label	types of cargo
Alternate Label	categories of freight
Hidden Label	Not Available
Description	Distinguish different types of cargo e.g. bulk cargo, liquid bulk cargo, and heavy materials.

Table 6: Examples of descriptive properties related to a concept

For each language, these descriptive properties differ in ontology. Table 7 gives the statistics of the descriptive properties for the English language, and Table 8 gives the same statistics for the German language.

Descriptive property	**Distinct Counts**
Preferred Label	14,529
Alternative Label	1,28,560
Hidden Label	1,097
Description	14,329

Table 7: Descriptive Statistics of the English language in the ontology

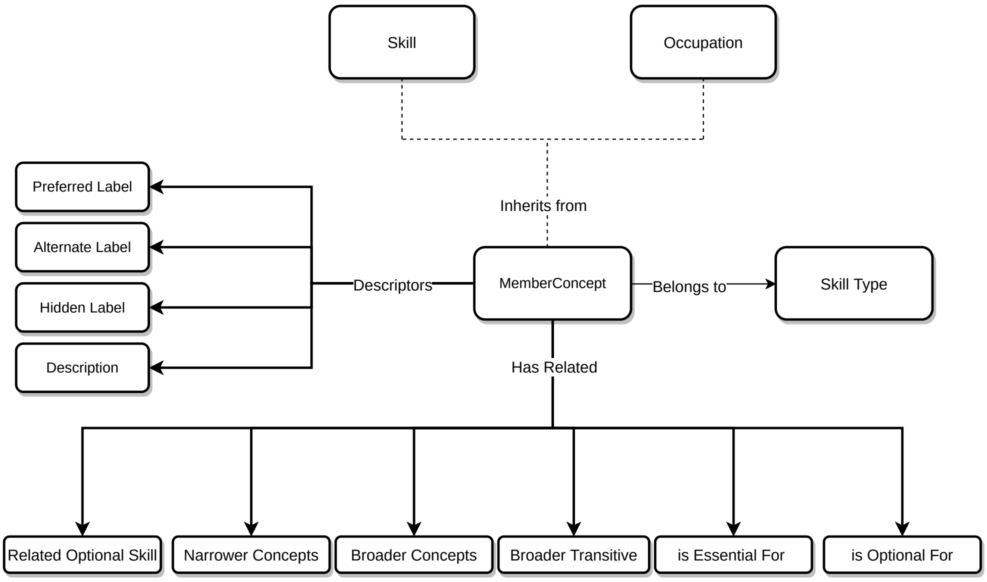
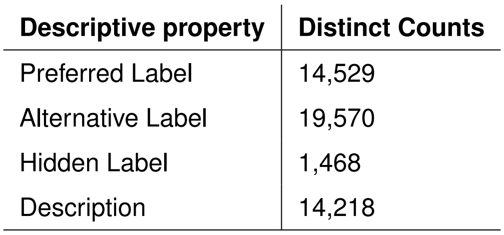


Figure 13: The simplified version of the Data Model of the ESCO ontology



Descriptive property	Distinct Counts
Preferred Label	14,529
Alternative Label	19,570
Hidden Label	1,468
Description	14,218

Table 8: Descriptive Statistics of the German language in the ontology

As we can see from the statistics of the descriptive properties above, the number of terms that are the same for both the English and the German language is only the preferred label for each concept URI. Thus we can infer that each concept in the ESCO ontology always has a preferred label, and this label is seen to be unique for each concept in the ontology in every language. We see that the description property is also available for most of the concept URIs. We can also see that for the English language, each concept URI has a lot of alternative labels attached to it, while in the case of the German language, the number of alternate labels is not so many. We can also note that the number of hidden labels is the least available descriptive statistic across all the available descriptive statistics of concept URIs in both languages.

The ESCO ontology also follows a multi-hierarchical structure for concepts. Multi-hierarchy comes when a single parent node has many child nodes coming out of it.

Since the ESCO ontology also has a multi-hierarchical structure, each concept has a parent class property and a child class property that can be used to encompass that particular concept. These classes help map concepts that might belong to more than one parent class and make it easier to keep track of the multi-hierarchy inside the ontology. The following properties define the hierarchies of a concept as per the skos vocabulary: narrower, broader, broaderTransitive, where each of the terms respectively describes the URIs of the entities that each concept contains, that each concept comes from, and the class of the parent from which the concept comes from. The statistics of the properties that define the hierarchy of each concept are shown in Table 9.

Hierarchical property	Unique URI counts	Unique Hierarchial property counts
Narrower classes	2,580	6,271
Broader classes	14,529	2,721
BroaderTransitive classes	14,529	2,851

Table 9: Statistics of the hierarchical properties of the ontology

From Table 9, we can see the counts of the hierarchical properties. We can see that for a total of 2,580 unique concept URIs, there are a total of 6,217 unique narrower classes available. This would imply that only some of the concepts have narrower classes, i.e., not all concepts contain a narrower class that shows more concepts falling under a concept. However, we can see from the broader class statistics that there are a total of 2,721 unique broader classes for all 14,529 concepts. This would imply that each broader class contains more than one concept inside it, thus implying that the data is multi-hierarchical. A similar multi-hierarchy structure can be seen for the broader transitive class in the ESCO ontology. A simplified version of the official ESCO data model, which represents all the above descriptive and hierarchical properties of the ontology, is visualized in Figure 13.

So to summarize, for each concept URI in the ESCO ontology, we have a set of URIs that are related to it as properties, some of which are descriptive and some of which are hierarchical properties with a different URI. All concepts have a preferred label that is unique to each concept, and most of the time the description of a concept is also available. Each concept has a broader class from which it is derived, and multiple concepts can have the same broader class to which they belong. This behavior induces a multi-hierarchical structure for concepts in the ESCO ontology.

3.3 Job Ads Data

Job ads data comprises terms in a job application that candidates use to apply for a job in a particular field. The fields that candidates apply to vary from a broad range including a general worker of any type to medicine and engineering. The job ads from which terms are extracted are German-speaking job ads in Switzerland, and thus all the terms that are extracted and we need to map are also in German. To be able to generalize the type of terms that appear in job ads from candidates of a different domain, we have created two types of datasets that contain the job ad terms from a wide domain of jobs available: one is a random sample that consists of 25 terms that frequently appear in job ads, and one is a challenge sample that contains 15 tailored challenge examples that are ambiguous for even the humans to annotate. Examples of the types of terms in the random and the challenge gold standard datasets are shown in Table 10.

Gold Standard Type	**Job-ad mentions**
Random	** *Feuerwehr* ** Sanität
Random	** Gewinnung von Neukunden ** *Akquise*
Challenge	** *EDV* ** DOS, Windows
Challenge	Arztsekretärin ** *MPA* **

Table 10: Examples of job-ad terms in random and gold standard datasets. The terms in italics are the main job-ad term for which the annotations are made, and the terms separated by '*' are the context words surrounding the job-ad term.

These gold standard datasets are the ones that were used in Gnehm et al. [2022a]. This paper used a transition-based NER[13] method from spaCy for detecting the skill mentions, which were categorized into fine-grained levels of education skills, experience skills, and language skills. From these extracted skill requirements, the random and the challenge gold standard datasets were created that are used for evaluation in this thesis.

These two datasets contain only the main job ad expression, and sometimes also contain the words that surround the job ad terms. We take these two datasets and extend the them by adding the suggestions given by our models for annotations. The words surrounding the job ad terms act as a context to the main job ad term and might help us better map a particular job ad to its respective term in the ontology. For making the task of classification extensible to general languages, we create the

[13] Named Entity Recognition: `https://en.wikipedia.org/wiki/Named-entity_recognition`

job ads random sample and the challenge sample in 2 languages: the original job ad expression in German, and the term translations of job ad expression in English. The translations of the German job ad terms are done manually, by keeping the translations as accurate as possible when translated into the English language.

This concludes the data section, which gives a detailed description of the ESCO ontology data, its statistics, and the data from the job ads that contain skill requirement mentions that we have to map onto the ontology.

4 Domain Adaptation Methods For Transformer LLMs

Since the domain data that we have is the ESCO ontology which is a KG as explained in Section 3.2, we have all our data as triplets in the form of (entities, relations, entities) format. We use this triplet data as the available domain data for pre-training and fine-tuning purposes of LLMs. We use three main methods for domain adaptation and fine-tuning of LLMs, namely, Masked Language Modelling (MLM) (Devlin et al. [2019]), Multiple Negative Ranking (MNR) Loss (Henderson et al. [2017]), and Classification task. The implementations of these methods are explained in the following sections.

As the base model to use as a benchmark for the end task and to further fine-tune for domain adaptation, we use the XLM-RoBERTa model[1] introduced in Conneau et al. [2020], and the XLM-RoBERTa model checkpoint adapted to the job-ads domain as per Gnehm et al. [2022b]. The main advantage of using the XLM-RoBERTa model is that it was pre-trained on a large multilingual dataset of 2.5TB of filtered CommonCrawl data, and thus we can use the same base model as a benchmark for both the English and the German language datasets because of its multilingual nature. The XLM-RoBERTa checkpoint adapted to the job-ads data is fine-tuned on the German job-ad terms from the job-ads data. The XLM-RoBERTa checkpoint that is fine-tuned on the job-ads data is also used as a benchmark to further fine-tune and adapt, as many of the terms in the job-ads data are also included in the ESCO ontology and thus this checkpoint acts as a good starting point for fine-tuning.

4.1 Masked Language Modelling (MLM)

Masked language modeling (MLM) is a semi-supervised language modeling task used for learning text representations for a particular domain. A semi-supervised learn-

[1] XLM-RoBERTa: `https://huggingface.co/xlm-roberta-base`

ing[2] method refers to a method for processing unlabelled data to obtain meaningful representations of the data. These representations can be learned from completely unlabelled data and the data requirements are also not too high.

The original MLM method that was in the BERT paper (Devlin et al. [2019]) followed the usage of a *[SEP]* token between two input sentences, with the masking of some words from each of the two sentences randomly with the *[MASK]* token. This mechanism from the original implementation is also depicted in Figure 14. The two input sentences may or may not be related to each other, and the accuracy of the model is computed only from the predictions of the words that are masked with the special mask token. The masking of the words is used such that the model is made to reproduce the masked word back via its training.

4.1.1 Why Use MLM for Domain Adaptation?

MLM is beneficial for domain adaptation as it helps pre-trained LLMs to handle previously unseen data from outside the domain it was originally trained with. The model is exposed to a wide range of contexts and sentence structures that it has not previously seen in its training and makes it learn meaningful representation by using contexts of the words masked with the special mask token. This process of reproducing the masked work back helps domain adaptation for the underlying LLM, and also helps it generate rich word embeddings for the domain of the input data.

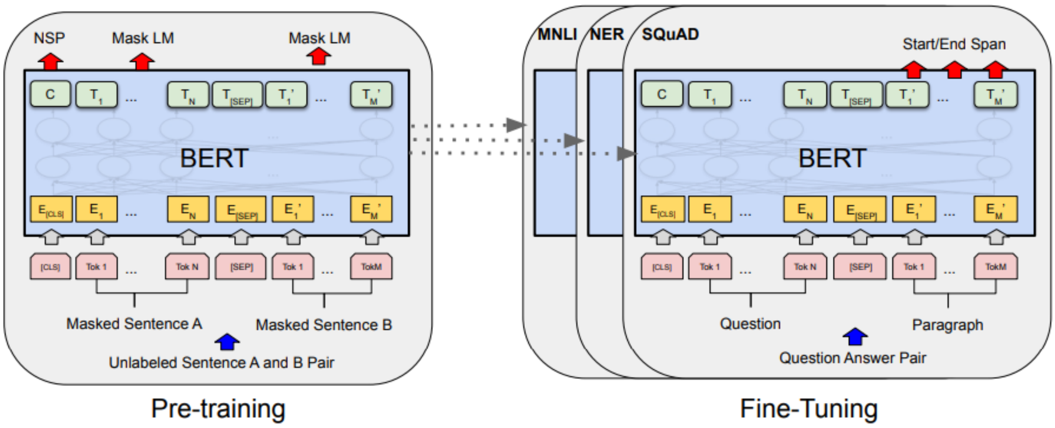


Figure 14: The pre-training and fine-tuning of the BERT transformer from the original paper. Image Source: Devlin et al. [2019]

[2] Semi-Supervised Learning: `https://en.wikipedia.org/wiki/Self-supervised_learning`

4.1.2 Implementation of MLM with ESCO Ontology Data

For our case, we use a slightly perturbed version of the vanilla MLM approach to further pre-train LLMs. As input sentences for MLM fine-tuning, we use a concept's *descriptions* from each of the English and the German language. concept descriptions are sentences that describe what the concept is about in a few sentences. We use the concept descriptions without any pairs, i.e., we only use a single input description of each concept without its concept URI as the input for the MLM model. The label of a concept is ignored because we do not gain any more knowledge by knowing if a particular description belongs to a particular concept for domain adaptation. All the descriptions passed thus act as a standalone corpus of sentences from the domain of the ESCO ontology, as we are interested in domain adaptation of the underlying LLM to the domain of ESCO ontology. Therefore, this corpus of descriptions of all of the concepts is used as the input for fine-tuning the underlying LLM via MLM. A few examples of the descriptions that are used for further pre-training the model are shown in Table 11.

Concept URI	**Description**
skill/bbeff2f6-4a3c-44af-8589-ac158f652d0d	Work with homeless individuals and support them with their needs, taking into consideration their vulnerability and isolation.
skill/3a08450a-7e60-449d-84cb-a39b3d8aaece	Provide recommendations for the installation and well functioning of hatcheries.
skill/c5f3e735-917b-4024-b2cf-19f4f75b2469	Livestock species and relevant genetics.
skill/6926e3fc-67ac-47ac-938e-4af566866ebd	Liaise with film exhibitors to persuade them to show the film or series.

Table 11: Examples of input data for MLM task. Only the sentences from the 'Description' column are passed to the model for fine-tuning.

From the above examples, we can see that different concepts have a different URI and a different description attached to them. Using the label of each unique concept, in this case, does not help the task of domain adaptation, as all such labels used would be true labels and there would be no false labels. Therefore, we only pass the descriptions associated with each concept for the fine-tuning task.

The parameters for the masking are set as per the original implementation, i.e., 15% of the tokens from the input descriptions were randomly masked. Of the 15%, 80% of the tokens were replaced with the special token *[MASK]*, 10% of the tokens were replaced by random tokens which are different from the original word being masked, and the remaining 10% of the times the tokens are left as is. The loss is only computed on the tokens that were masked with the special *[MASK]* token, and the random token replacement trains the model to generate representations of tokens at each position.

Also, since transformer models need the maximum input sentence to be 512 subtokens and it is possible that the input descriptions of the concepts are longer than 512 subtokens, we have used a strategy to create chunks of the input sentences such that each chunk is 512 subtokens long. The strategy involves cutting single descriptions into padding chunks, with each chunk containing 512 subtokens.

4.1.3 Validation Dataset for MLM

As part of the validation set to check if the fine-tuning itself improves the model performance, we selected 10% of all the available concept descriptions as the validation set. On this validation set, the evaluation loss is computed on the special *[MASK]* tokens while the model is fine-tuned. The loss is computed by checking if the word masked by special *[MASK]* token is predicted correctly by the model or not, while the tokens generated by all the other words are ignored.

The MLM fine-tuning is used as a starting point for domain adaptation since it helps the model learn unseen contexts from the new domain used for fine-tuning. For the MLM fine-tuning, we have 2 models to use as starting points for fine-tuning, and thus we get a total of 2 fine-tuning variants for each language. The results of the MLM model are stored with the tag *'xlm-oob-ft'* and *'xlm-jobads-ft'* for the default XLM-RoBERTa model and the job-ads domain adapted XLM-RoBERTa model checkpoint used as a starting point, respectively.

4.2 Multiple Negative Ranking (MNR) Loss

Multiple Negative Ranking (MNR) is a technique used for training models such that they get better at handling imbalanced class distributions. The Multiple Negative Ranking Loss (Henderson et al. [2017]) is a loss function that employs the MNR technique. It is an unsupervised learning method[3], that takes pairs of positively and negatively related sentences to an anchor and optimizes their distances. More precisely, the main objective of the MNR loss function is minimizing the vector distances between the input anchor and the positive sentence pair, and simultaneously maximizing the distance between the input anchor and all other pairs that are not similar to it. In short, all positive pairs should be brought closer to each other and all unrelated pairs should be pushed further away from each other. MNR loss is an unsupervised loss, as all the sentence pairs, both positive and negative, that we pass for the fine-tuning task are unlabelled. Sentences are only paired by their relationship to the anchor sentence as either being a positively related sentence or a negatively related sentence. A depiction of this is shown in Figure 15.

The training for MNR Loss can be done in either sentence pairs or triplets, wherein the anchor is always fixed and sentences are paired with it. For the case of triplets, the data consists of the anchor term paired with one positively related term and one negatively related term. In the case of triplet data, the vector distances are again optimized such that the distances of the anchor and the positive pair are minimized and the distances of the anchor and the negative pair are maximized simultaneously. For the case of triplet data where we have an anchor and a negative pair, we can also give specific hard negatives so that the learning objective is more challenging for the model to solve from the fine-tuning.

4.2.1 Why Use MNR Loss for Fine-tuning?

Unlike traditional classification methods, MNR loss solves the objective by minimizing the distance between sentences that are positively related to an anchor and simultaneously maximizing the distance between the anchor and negatively related pairs. This approach is very beneficial for tasks of information retrieval, where there may be a large number of possible answers available for a question from the corpus, but only a smaller subset is relevant for the input query. The MNR loss objective helps the model rank positive pairs much higher than negative pairs, increasing the likelihood of the relevance of the most positive answer. Moreover, this loss function directly influences the vectors of the sentence pairs and is thus more direct in helping the model understand the input domain data. This is why MNR loss is beneficial for fine-tuning to help the model return the most relevant answer from the set of available answers for an input query.

[3] Unsupervised Learning: `https://mitpress.mit.edu/9780262029445/fundamentals-of-machine-learning-for-predictive-data-analytics/`

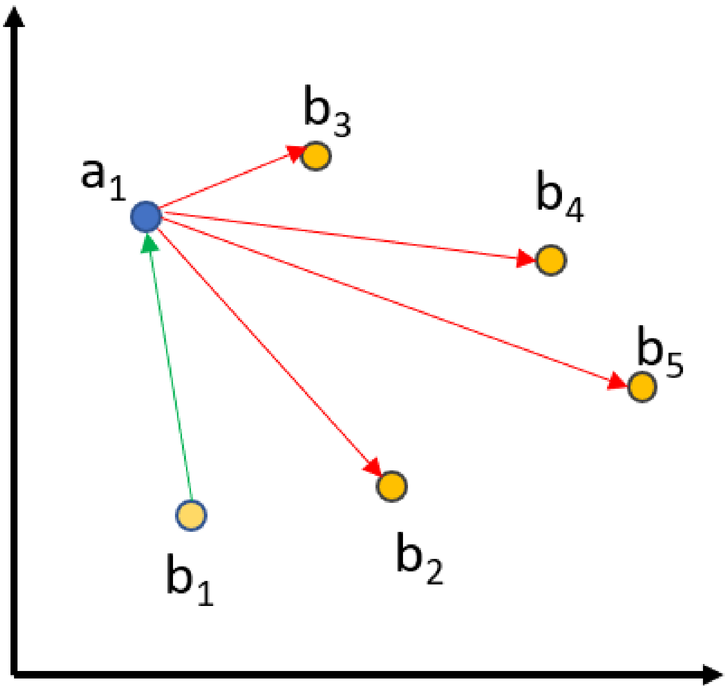


Figure 15: The objective of the MNR loss. The sentence vector pair ‘a1-b1’ are related and thus their distance is minimized, and all the other sentence pairs are unrelated and thus their distances are maximized. Image Source: `https://sbert.net/examples/training/nli/README.html`

4.2.2 Implementation of MNR with ESCO Ontology Data

For our fine-tuning of LLMs via MNR loss, we created triplet data from the ontology, with one of concept’s descriptive property as being the anchor, another descriptive property as being the positive pair, and all of the unrelated concept’s descriptive properties as being the negative pairs. The anchor itself was always the concept’s preferred label, while it was paired with either a different type of label or the description of the positive or negative classes. In general, the triplet pairs consisted of either all preferred labels of anchor, positive and negative class, or preferred label of the anchor and descriptions of the positive and the negative classes.

We classified negatives as all the concepts that do not belong to either the *Narrower Class* or the *Broader Class* of the input concept as shown in Figure 13. The Narrower class contains all the concept URIs that fall under a particular concept, and the Broader Class contains all the concept URIs that a particular concept comes from. The Narrower and the Broader classes can also be thought of as the child and the parent concepts, respectively. Thus, for each concept, we have a vast number of negative concepts available across the ontology. However, we need to make sure that the ratio of the positive and the negative pairs are kept to a similar level in the training data, otherwise, the negative samples would always make the dataset unbalanced. For this reason, we mined specific hard negatives from the set of all the available negatives and paired them with anchors.

4.2.2.1 Hard Negative Mining

To select negatives that make solving the problem with MNR loss more challenging, we mined hard negatives that are lexically similar to the input anchor concept term but are still outside the *Narrower* and the *Broader* classes of the input concept. This approach is largely inspired by Li et al. [2019], wherein they use semi-hard sampling with a base model to compute hard negatives. Negative concept's descriptive properties, either the label or the description, that contain words other than stop words from the anchor concept's descriptive properties are first selected, and then the nearest neighbor is computed with out-of-the-box models for the anchor property, and the available negative concept property. From these nearest neighbors, concepts that have the highest level of intersection with the anchor class are taken to be the hard negative for that particular anchor. The flow diagram to compute the hard negatives for an input anchor class is shown in Figure 16.

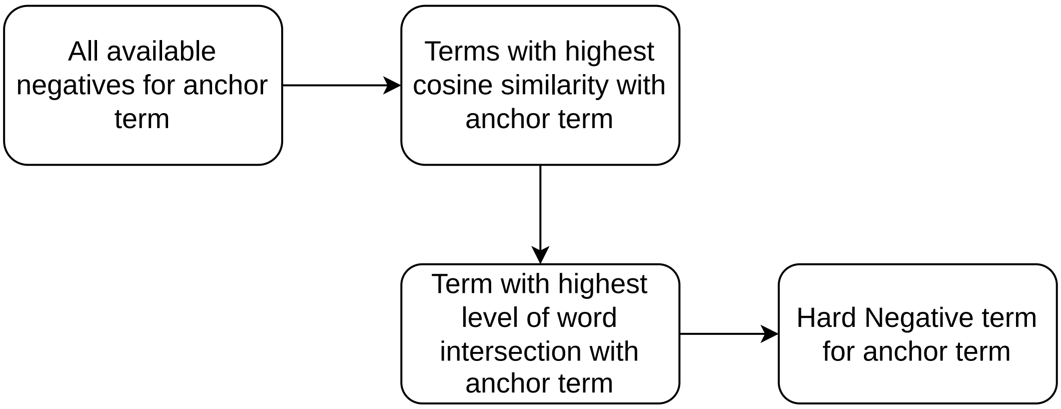

All available negatives for anchor term
Terms with highest cosine similarity with anchor term
Term with highest level of word intersection with anchor term
Hard Negative term for anchor term

Figure 16: Flow diagram to compute hard negatives for each anchor term.

To compute the nearest neighbors, two different models are used for the English and the German language. For English, we use the all-mpnet-base-v2[4] model, and

[4] all-mpnet-base-v2: `https://huggingface.co/sentence-transformers/all-mpnet-base-v2`

for German we use the paraphrase-multilingual-MiniLM-L12-v2[5] model. These two models are used out-of-the-box to compute which of all the negative terms are most similar to the anchor term as a preliminary step, and only after this step do we choose the term with the highest number of words intersecting with the anchor term as the hard negative for that particular anchor.

Thus we have two variants of triplet data to feed to the model for the MNR training: one with randomly sampled negatives and one with mined hard negatives. The examples of input data from each of these two samples are shown in Table 12.

Negative Sample Type	Anchor	Positive Term	Negative Term
Random Sample	manage musical staff	manage staff of music	apply psychoeducation
Hard Sample	manage musical staff	manage staff of music	*manage* quality systems
Random Sample	supervise correctional procedures	oversee prison procedures	manage clients' money matter
Hard Sample	supervise correctional procedures	oversee prison procedures	*supervise* governing warehouse safety *procedures*
Random Sample	identify available services	determine rehabilitation services	ensure the quality of aircraft systems installation
Hard Sample	identify available services	determine rehabilitation services	*services* provided by railway companies

Table 12: Examples of input data for MNR task. Both hard and randomly sampled negative datasets are shown. The words in italics are the ones that intersect with the anchor term in Hard Sample negative data.

As we can see, the hard negative sampled triplet data contains words appearing in both the anchor term and the negative term. This would impose a harder challenge on the model to try and maximize the distance between the anchor term and the negative term, as simply using distances at the word level would be ruled out with the implementation of hard negative samples. The model would instead have to semantically construct sentence embeddings that help it maximize the distance between the anchor term and the negative term.

For the MNR loss training, all the hyper-parameters are set to the default parameters as mentioned in Henderson et al. [2017].

4.2.3 Validation Dataset for MNR Loss

To compare the difference in the learning that randomly sampled negatives and hard negatives variants achieve and also the improvement of the model as the fine-tuning happens, we created a static validation dataset. This validation dataset was created

[5] multilingual-MiniLM: `https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2`

using 10% of data from the hard negative triplets by selecting the second-best hard negative for each of the input anchor terms. As benchmarks to perform fine-tuning and further improve, the 2 XLM-RoBERTa checkpoints, one out-of-the-box and one domain adapted on job-ads data were used. Along with this checkpoint, the checkpoints fine-tuned via MLM in Section 4.1 were also used. The models fine-tuned via the MNR strategy were prefixed with the term *'mnr'*, and an example of a model that used MLM checkpoint as a base, with hard negative triplet data, and was further fine-tuned was saved as *'mnr-hardneg_xlm-jobads-ft'*. Thus, a total of 4 model variants are available for the MNR fine-tuning for each language per triplet dataset. However, since we are using two variants of triplet data, one with randomly sampled negatives and one with hard sampled negatives, we get a total of 8 variants of MNR fine-tuning per language.

4.3 Classification Task

In NLP, classification tasks are formulated as supervised learning tasks[6], wherein the task involves assigning predefined labels to a set of input sentences. Usually, classification tasks are used for solving problems like sentiment detection, language detection, etc. where we can have a set of pre-defined labels to input sentences and train a model, later asking it to classify sentences into one of the pre-defined categories. The task of classification is solved by learning patterns from the input sentences that relate to the label that is provided to the sentences.

For fine-tuning LLMs via the classification task, we use the classification head that only uses the *[CLS]* token. This classification head then takes a set of sentences along with the label and creates a feature space that is later used to classify a new set of sentence pairs. For our case, as a label, an artificial binary label was created for sentence pairs, based on whether or not the passed input sentences were related to each other or not. The label creation is done similarly as it is done in the Natural Language Inference (NLI)[7] task, wherein we take two sentences as input premise and the labels that are given to them act as a hypothesis in classifying whether those sentences are either related, unrelated, or neutral to each other.

[6] Supervised Learning: `https://en.wikipedia.org/wiki/Supervised_learning`

[7] Natural Language Inference: `https://sbert.net/examples/training/nli/README.html`

4.3.1 Why is the Classification Task Useful for Domain Adaptation?

Domain adaptation refers to adapting a model pre-trained on one domain to be able to work on another domain. For this task, classification is often helpful by fine-tuning the pre-trained model to the target domain by training a classifier. By training such a classifier, the model can learn to identify features and patterns from the target domain that help it perform a better classification of the target domain data. This is why classification as a domain adaptation task is useful for adapting a pre-trained model to a different target domain.

4.3.2 Implementation of Classification Task with ESCO Ontology Data

For the classification task, as two input sentences, two attributes of each concept are paired together. These attributes could either be the concept label, like its preferred label or its alternate label, or the concept description describing what a particular concept is about. While creating the term pairs from the descriptive attributes of a concept, we also randomly shuffle the attributes for a particular concept, thus the sentence pairs created could either be term-description or description-term pairs. A corresponding binary label is created for the pair depending on the attributes of the selected concept. For attributes that are related to each other, i.e., for a positive sentence pair, a label *'0'* is assigned. Similarly, for unrelated sentences, a label *'1'* is assigned. The sentence pairs that are unrelated to each other are generated via the same strategy of hard negative mining as mentioned in Section 4.2.2.1.

For an input concept, negative concepts are first selected based on the exclusion from the Narrower and Broader class of the input concept, and then the concept properties that had the input term present in them were selected as negatives. From this set of negatives, the concepts which had the highest cosine similarity score with the input term were taken to be as hard negatives for the input term. Thus, the attribute of the unrelated pair contains words from the concept label but is from a family outside the *Narrower* and the *Broader* class of the concept. These artificially created labels, along with both positive and negative sentence attribute pairs, are fed into a transformer model with a classification head, and the accuracy score is computed to check if the input LLM is able to learn from the classification task. The accuracy score is computed from the artificially generated labels that we provide for each pair of sentences based on whether the model is able to predict the same label for the sentence pairs as the classifier is trained.

For our case, as input features for each label, a particular concept's label is taken and paired with a different attribute of the concept, either label or description. Thus, the input data for the classification task is of the form label-label or label-description or description-label in our case. The input sentences are attached to each other with the *[SEP]* special token to indicate separation between the concept attributes passed. The label of each pair of input sentences is used as an identifier to determine whether or not the two sentences passed are related to each other or not.

The examples of the input data for the classification task with the artificial labels are shown in Table 13.

Anchor Term	**Descriptive Term**	**Label**
teach housekeeping skills	teach domestic science skills	0
teach housekeeping skills	*teach* archeology	1
apply credit risk policy	implement credit risk policy	0
apply credit risk policy	*apply* pre-stitching techniques	1

Table 13: Examples of input data for the Classification task. The labels '0' represent a positive pair, and the label '1' represents negative sentence pairs. Words in italics are the words that intersect with the anchor term for negative pairs.

4.3.3 Validation Dataset for Classification

For the classification task, the main data consisted of sentence pairs with artificially generated binary labels to them, indicating if the sentence pairs are related to each other or not. As a validation dataset to check the improvements while training the classifier, 10% of the total data was used as a held-out validation dataset. The accuracy was computed over the labels of this validation set while training to check how the performance of the classifier changed during the training phase.

As base models for classification to further fine-tune, we use the out-of-the-box XLM-RoBERTa model and the XLM-RoBERTa model fine-tuned on job-ads data, and model checkpoints from the MLM fine-tuning task. Thus the classification tasks were also further build on top of different model checkpoints. A total of 4 model variants are created by fine-tuning via the classification task for each language.

Alongside the traditional methods of fine-tuning an LLM[8] by using all the data from the domain, there are also few shot-learning methods. Few-shot learning methods are a good alternative in scenarios wherein labeling all the available training data is time-consuming. We could also use few shot learning without prompts (Tunstall et al. [2022]), which first fine-tunes a pre-trained LLM on a small amount of domain data, and then uses it to generate rich word embeddings.

This concludes the section on describing the methods used to fine-tune and domain-adapt the LLMs to the ESCO ontology.

[8] Large Language Models: `https://huggingface.co/blog/large-language-models`

5 Experiment Pipeline

Since there were a lot of variants of models that were to be trained via the fine-tuning tasks, it is helpful if a proper process is followed for the experiments. Experimental pipelines that are controlled by configurations such that faster iterations with different model checkpoints are possible are beneficial for the type of work done in this thesis. This section outlines the overall pipeline setup for conducting the fine-tuning experiments described in Chapter 4.

5.1 Data Extraction and Storage

As mentioned in Section 3.2, the ESCO ontology data was available in an RDF turtle file format. This file contains all data in the form of subject, predicate, and object triplet pairs, and can be thought of as a very large text file containing triplets of the above form. For extracting data from the turtle file, we made use of the SPARQL[1] query language for RDF. This query language is similar to structure to the Structured Query Language (SQL)[2]. The main difference that SPARQL has with SQL is that SPARQL needs a set of vocabulary from the ontology to operate on. Recall that the vocabularies used in the ESCO ontology are highlighted in Section 3.2.5.

For extracting data from each of the descriptive and hierarchical properties of the ontology, we wrote different SPARQL queries. Each SPARQL query dealt with extracting one of the related properties of the Concept as outlined in Figure 13. The RDFLib package[3] in Python[4] was used to load the turtle file into memory and execute SPARQL queries to extract data.

Since most of the data consisted of the URI of a Concept and the textual description of the property used in the SPARQL query, the data was stored in tab-separated

[1] SPARQL: `https://www.w3.org/TR/rdf-sparql-query/`

[2] SQL: `https://en.wikipedia.org/wiki/SQL`

[3] RDFLib: `https://rdflib.readthedocs.io/en/stable/`

[4] Python: `https://www.python.org/`

values (TSV)[5] files. These files were extracted only once for all the relationships and hierarchies of a Concept and stored locally. These files were the main data files that were used for fine-tuning tasks.

5.2 Programming Method for Fine-tuning Scripts

The central programming language used for creating the pipelines for fine-tuning was Python[6]. All the scripts related to fine-tuning, checkpointing, and storage were written in Python. In general, instead of creating many scripts that contain the exact parameters used for fine-tuning, an approach to creating skeleton scripts was followed. The parameters that were to be used by a particular fine-tuning script were later passed as a JSON[7] configuration file. This made it much easier to have only one skeleton script for one type of fine-tuning method, and later create many configuration files to pass parameters to the skeleton script. This helped maintain the code base, making it easier to iterate in parallel and keep track of different experiments starting and finishing simultaneously.

5.3 NLP Ecosystem Used

Since most of the time we were dealing with transformer models and LLMs to use as a base for fine-tuning, the main resource used to get the model checkpoints and model parameters was HuggingFace[8]. HuggingFace is an open-source AI community, largely dealing with open-sourcing transformer models for general usage by the community. The main benchmark model used by us, the XLM-RoBERTa-base[9] model, was fetched from the HuggingFace models page.

Another main source of getting resources for fine-tuning was the SentenceTransformers framework[10]. This open-source framework contains a collection of models and methods that make the usage of sentence transformers for fine-tuning and domain adaptation much easier. The main implementation of MNR loss was taken from this framework.

[5] TSV file: https://en.wikipedia.org/wiki/Tab-separated_values

[6] Python: https://www.python.org/

[7] JSON: https://www.json.org/json-en.html

[8] HuggingFace: https://huggingface.co/

[9] XlM-RoBERTa-base: https://huggingface.co/xlm-roberta-base

[10] SentenceTransformers: https://www.sbert.net/

As part of the logging strategy to log the results of the validation data, Tensorboard[11] was used. With tensorboard, all the evaluation, loss, and accuracy curves of the training procedure were logged so that they are available for analysis at a later stage. Many of the curves from tensorboard are also shown in the Section 6.1 on the internal evaluation of the models.

5.4 Data Creation After Model Fine-tuning

Once the variants mentioned in Chapter 4 are fine-tuned and suitable checkpoints are obtained, then the data creation for the end task of classifying job-ad terms onto the ESCO ontology is conducted. As part of this data creation, from each of the two files for evaluation, the random sample job-ad term file and the challenge sample job-ad term file, we compute the euclidean distances of the job-ad term with the concepts in the ESCO ontology. In other words, for every term which is present in the 2 evaluation files, their top 20 nearest terms from the ESCO ontology are computed using each of the fine-tuned model checkpoints by checking the euclidean distances of each input term with the terms present in the ESCO ontology.

For computing the top 20 nearest terms, the job-ad terms are taken both with and without the available context. The context in this case is the extra terms that are provided along with the job-ad terms, as prefix and suffix words. The resulting nearest terms are different when the extra context words are used along with the job-ad terms, and thus it is used to check if the predictions with and without context make any significant difference on the end task of classification on the ESCO ontology.

This concludes the section describing the creation of the pipeline for fine-tuning tasks for domain adaptation.

[11] Tensorboard: `https://www.tensorflow.org/tensorboard`

6 Results

As many variants of models were trained as part of the fine-tuning and domain adaptation, there were many different types of metrics being logged from each of the fine-tuning runs. Many of these metrics were related to describing the internal performance of the models during the fine-tuning step. This internal performance that each model gets is different from the end task of classifying the job-ad term onto the ESCO ontology, rather the model metrics logged during fine-tuning only explained how the fine-tuning of the model influenced its performance over the held-out validation set over time.

Because of this, the results section is split into two subsections: one section explains the internal evaluation done on the models to check if the models are learning from the fine-tuning tasks, and the external evaluation is done on the end task of classification of job-ad terms onto the ESCO ontology terms.

6.1 Internal Evaluation of Models

For each of the fine-tuning tasks for domain adaptation described in Chapter 4, we have conducted internal evaluations. The means of this internal evaluation is to check if the fine-tuning in itself is helping the model improve on the validation dataset or not. While the improvement in the internal evaluation does not necessarily imply an improvement in the external evaluation, the internal evaluation is still necessary to check if the tasks themselves bring about any improvements via fine-tuning.

6.1.0.1 Evaluation Curve Interpretation

Each evaluation curve includes a legend that corresponds to the type of model checkpoint used as the initial point for the domain adaptation task that the curve represents. The legend is presented alongside the curve. The available options for the legend on the internal evaluation curves are summarized in Table 14.

Legend Name	Model represented
xlm-oob	XLM-RoBERTa out-of-the-box checkpoint
xlm-jobads	XLM-RoBERTa checkpoint adapted on job-ads domain
xlm-oob-ft	XLM-RoBERTa out-of-the-box checkpoint fine-tuned via MLM
xlm-jobads-ft	XLM-RoBERTa checkpoint adapted on job-ads domain and fine-tuned via MLM

Table 14: Table representing all the possible values of the model starting points for the domain adaptation tasks

6.1.1 MLM Evaluation

For the MLM task, we used 2 variants of models as benchmarks, the out-of-the-box XLM-RoBERTa model and the XLM-RoBERTa adapted to the job-ads domain created as per the Paper Gnehm et al. [2022b]. As mentioned in Section 4.1.3, 10% of the input sentences that contain descriptions of Concepts from the ontology were taken as the validation dataset for the MLM task for both English and German languages separately. The evaluation loss is computed by checking whether the masked token was predicted to be the original word. Based on this computation of the loss, the curves of the evaluation loss on the validation dataset for the English language are shown in Figure 17 and for the German language in Figure 18.

In Figure 17, the red curve on the bottom is the evaluation loss curve for the job-ads checkpoint during training, and the blue curve on top is the loss curve for the out-of-the-box XLM-RoBERTa checkpoint. The loss curve shows that the evaluation loss for the job-ads checkpoint is lower after 2k training steps when compared to the evaluation loss of the out-of-the-box XLM-RoBERTa model. A training step in this curve consists of one update made after seeing a batch of the input sentences, where the batch is a hyperparameter set by us during the training phase.

A hypothesis for this result is valid is that the XLM-RoBERTa checkpoint adapted to the job-ads domain has already seen many terms from the job-ads, and these terms are also related to the words appearing in the descriptions of the Concepts. The job-ads data consist of terms in the German language that applicants use in their applications to apply for jobs. For both model variants, the loss initially starts at a higher value and seems to be flattening around 2k training steps, which is approximately 6 epochs of the whole training data.

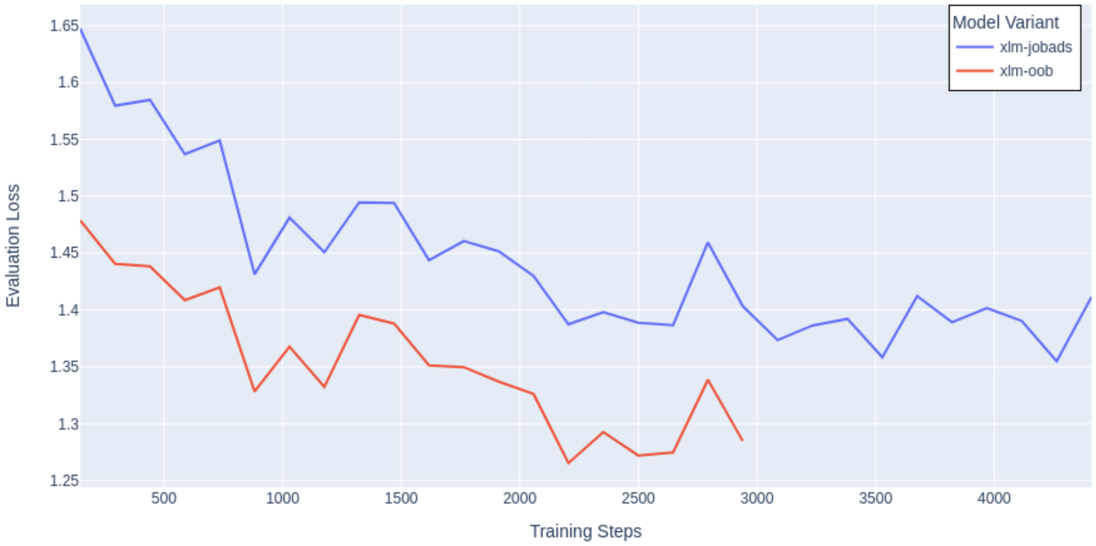


Figure 17: Evaluation Losses of MLM for the English data. The bottom line is from the job-ads checkpoint, and the top line is from the XLM-RoBERTa checkpoint.

In the Figure 18, the red curve on the bottom is the evaluation loss curve for the job-ads checkpoint during training, and the blue curve on top is the loss curve for the out-of-the-box XLM-RoBERTa checkpoint. The loss curve shows that the evaluation loss for the job-ads checkpoint and the out-of-the-box XLM-RoBERTa model is flattening after 2.5k training steps. Moreover, it can be seen that the overall loss values of the XLM-RoBERTa checkpoint adapted to the job-ads domain are lower for the German language datasets in comparison to the English language datasets as seen in Figure 17. This is the job-ads data on which the XLM-RoBERTa checkpoint is adapted are all in the German language, and thus the MLM task on the German language benefits from this. A training step in this curve consists of one update made after seeing a batch of the input sentences, where the batch size is a hyperparameter set by us during the training phase.

We can make the same inference as before for the German language dataset since we are utilizing the job-ads checkpoint, which has already undergone fine-tuning on job-ads domain data, as a base model for MLM on the German datasets.

The dissimilarity in the evaluation curves depicted in Figure 17 and Figure 18 can be attributed to the difference in the number of epochs for which the fine-tuning scripts

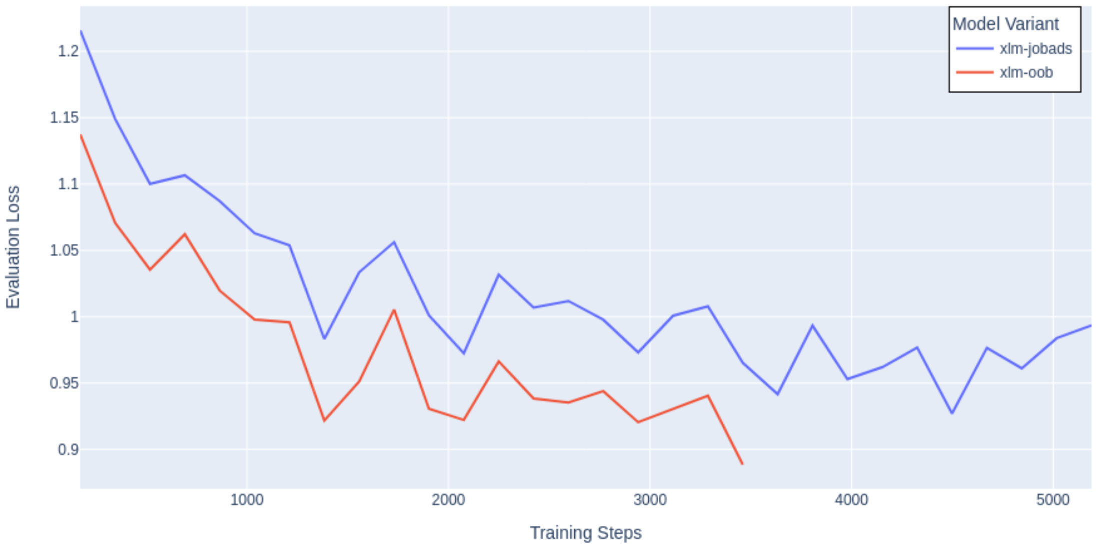


Figure 18: Evaluation Losses of MLM for the German data. The bottom line is from the job-ads checkpoint, and the top line is from the XLM-RoBERTa checkpoint

were executed. Additionally, there is a difference in the total number of steps carried out by the fine-tuning models due to the disparity in the inherent corpus containing descriptions of the Concepts for the English and German languages. Specifically, the German language corpus of input descriptions is smaller compared to the English language corpus, as indicated by the statistics in Table 7 and Table 8.

Based on the assessment of MLM fine-tuning for the English and German datasets, it is evident that this approach enhances the ability to predict masked words according to the ESCO ontology domain. The decrease in evaluation loss signifies that the model can accurately predict the masked words in their original context. As the original data pertains to Concept descriptions from the ESCO ontology, the model has learned to predict words in a manner consistent with the ESCO ontology. In addition to evaluation loss, the perplexity score after MLM fine-tuning also indicates a comparable outcome, as it is also calculated based on the model's ability to predict masked words. Still, the evaluation loss gives also a similar result as it is computed by the model predicting the masked words back.

Therefore, it can be inferred that MLM fine-tuning enables the model to acclimate to the ESCO ontology domain. Since this objective aligns with the end task of categorizing job-ad terms onto the ESCO ontology, it is reasonable to anticipate that the MLM fine-tuned model checkpoints would serve as a reliable foundation for subsequent fine-tuning and adjustment.

6.1.2 MNR Evaluation

To evaluate the performance of the MNR task, we employed four model variants as initial models: the two fine-tuned checkpoints obtained from the MLM task, the out-of-the-box XLM-RoBERTa model, and the XLM-RoBERTa checkpoint modified for the job-ads domain. These four variants are applied based on the language of the triplet dataset for the MNR fine-tuning, and there are two types of triplet datasets in total. As described in Section 4.2.3, we utilized 10% of the hard negative triplet data as the static validation dataset for all MNR fine-tuning variants.

The accuracy plots are utilized in the MNR evaluations to depict the impact of training. These accuracy scores are calculated by determining the count of triplets where the cosine similarity between the anchor and positive term is lesser than the distance between the anchor and negative term, divided by the total number of available triplets. Therefore, higher accuracy indicates a larger number of triplets with a smaller cosine similarity distance between the anchor and positive pair, as compared to the anchor and negative pair.

6.1.2.1 MNR Evaluation Results for English Dataset

The accuracy plots of the randomly sampled negatives and the hard sampled negatives of the English dataset are shown in Figure 19 and Figure 20.

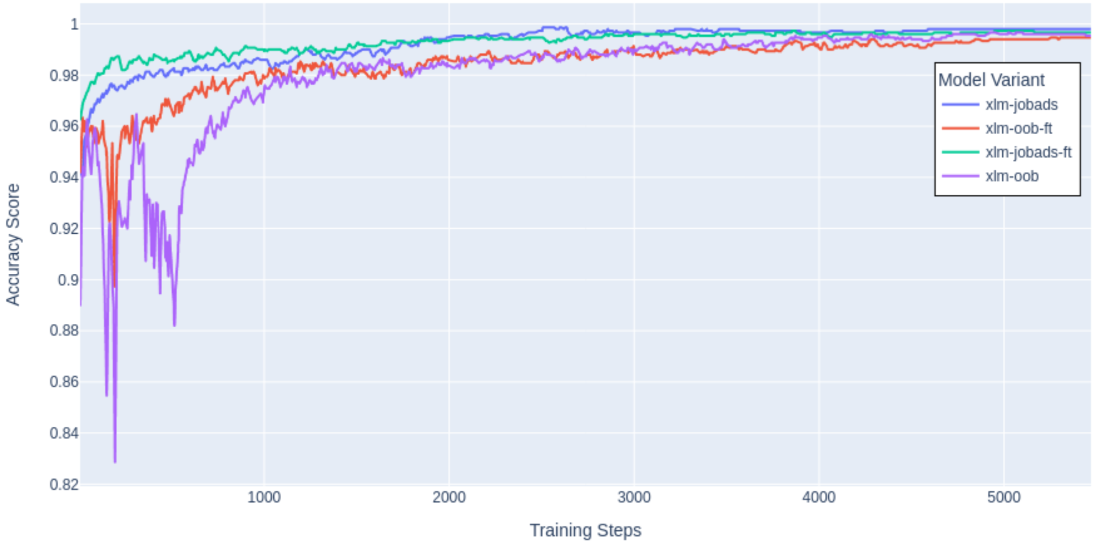


Figure 19: Plots of the accuracy scores of the MNR fine-tuning for the English dataset with hard sampled negatives

From the Figure 19, we can see a summary of the models that are used in the MNR fine-tuning task with the hard negative triplet dataset in the Table 15.

Color of the Curve	Model starting point for MNR training
Green	XLM-RoBERTa checkpoint, adapted on job-ads checkpoint and fine-tuned via MLM
Blue	XLM-RoBERTa checkpoint, adapted on job-ads checkpoint
Orange	XLM-RoBERTa out-of-the-box checkpoint, previously fine-tuned via MLM
Purple	XLM-RoBERTa out-of-the-box checkpoint

Table 15: Descriptions of accuracy plots in Figure 19 of hard negative triplet data for MNR fine-tuning in English

From the Figure 19, we can see that in the initial stages of training, the accuracy scores are low. Over time as the training continues, the accuracy score gradually increases, indicating an improvement in the model's ability to differentiate between positive and negative pairs. The fluctuations in accuracy are expected during the training process and may indicate overfitting or a lack of diverse training data. Towards the end of the training process, the accuracy score stabilizes, indicating that the model has reached a point where further training is unlikely to yield significant improvements.

Despite all four variants achieving accuracy scores close to 99%, the out-of-the-box XLM-RoBERTa and the MLM fine-tuned out-of-the-box XLM-RoBERTa models exhibit poor performance in the early stages of training. This finding suggests that using the out-of-the-box XLM-RoBERTa model or its MLM fine-tuned variant on its own may not be sufficient for zero-shot learning applications. However, the results show that further adaptation of these models can increase their accuracy in correctly classifying positive pairs compared to negative pairs.

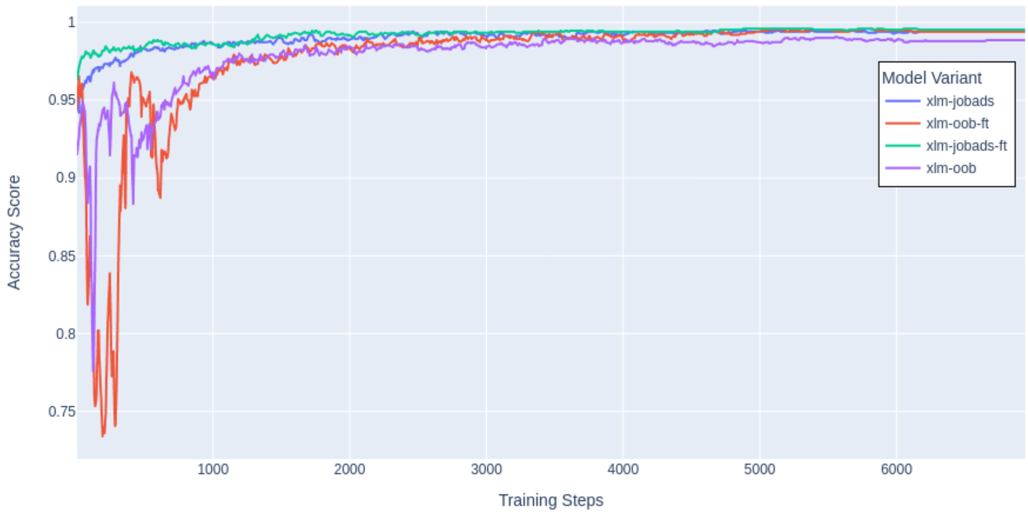


Figure 20: Plots of the accuracy scores of the MNR fine-tuning for the English dataset with randomly sampled negatives

The Figure 20 shows the accuracy plots of the MNR training with the randomly sampled negatives, where the summary of the models used as a starting point for the fine-tuning is given in Table 16.

Color of the Curve	Model starting point for MNR training
Green	XLM-RoBERTa checkpoint, adapted on job-ads checkpoint and fine-tuned via MLM
Blue	XLM-RoBERTa checkpoint, adapted on job-ads checkpoint
Purple	XLM-RoBERTa out-of-the-box checkpoint, previously fine-tuned via MLM
Orange	XLM-RoBERTa out-of-the-box checkpoint

Table 16: Descriptions of accuracy plots in Figure 20 of random sampled negative MNR fine-tuning in English

We can see from the Figure 20 that all of the 4 MNR variants reach the final accuracy of approximately 98% as the fine-tuning continues. However, the out-of-the-box XLM-RoBERTa model variant and the MLM fine-tuned variant of the XLM-RoBERTa model do not have higher accuracy in the earlier stages of fine-tuning. This behavior is similar to the behavior seen in the accuracy plots in Figure 19. This would again imply that the out-of-the-box variant of the XLM-RoBERTa model and the MLM fine-tuned variant of the XLM-RoBERTa model are not good to be used in the zero-shot setting.

Based on the accuracy plots presented in Figure 19 and Figure 20, it can be inferred that all the MNR fine-tuned models, using both randomly and hard sampled negatives, improve in accuracy with continued fine-tuning. Additionally, the out-of-the-box XLM-RoBERTa model and its MLM fine-tuned variant are not suitable for zero-shot predictions on the triplet data. Also, the final accuracy that the XLM-RoBERTa variant gains with the randomly sampled negative dataset is 1pp lower than the final accuracy of the same model with the hard negative sampled dataset. This is due to the fact that the evaluation data that we use for the MNR fine-tuning task is a subset of the hard negative triplet dataset.

6.1.2.2 MNR Evaluation Results for German Dataset

For the MNR fine-tuning with the German datasets, we can see that the accuracy score plots in Figure 21 and Figure 22.

The Figure 21 contains four accuracy plots for the MNR fine-tuning on the German dataset with hard sampled negatives, with a summary of the models used as a starting point shown in Table 17.

Color of the Curve	**Model starting point for MNR training**
Green	XLM-RoBERTa checkpoint, adapted on job-ads checkpoint and fine-tuned via MLM
Blue	XLM-RoBERTa checkpoint, adapted on job-ads checkpoint
Orange	XLM-RoBERTa out-of-the-box checkpoint, initially fine-tuned via MLM
Purple	XLM-RoBERTa out-of-the-box checkpoint

Table 17: Descriptions of accuracy plots in Figure 21 of hard negative triplet data for MNR fine-tuning for German dataset

The accuracy plots in Figure 21 show that the XLM-RoBERTa checkpoints that are adapted to the job-ads data and initially fine-tuned via MLM achieve the highest accuracy after fine-tuning, reaching close to 99%. The out-of-the-box XLM-RoBERTa checkpoint has the lowest accuracy at around 92%, while the out-of-the-box check-

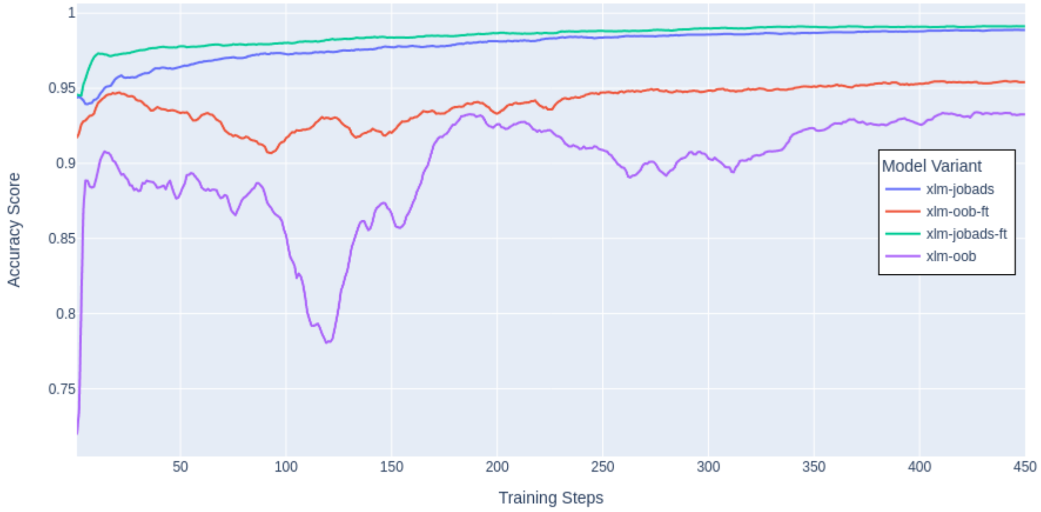


Figure 21: Plots of the accuracy scores of the MNR fine-tuning for the German dataset with hard sampled negatives

point initially fine-tuned via MLM has an accuracy of around 95%, which is still lower than the top-performing models. These results suggest that XLM-RoBERTa checkpoint domain-adapted on the job-ads data and fine-tuned via MLM on the ESCO ontology data are the most effective for the MNR fine-tuning of hard negative sampled triplet data in the German language.

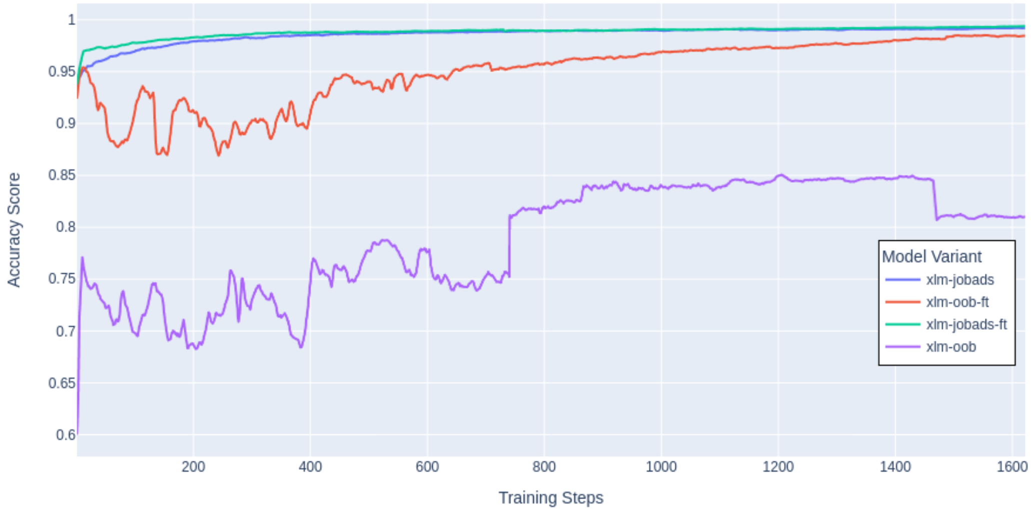


Figure 22: Plots of the accuracy scores of the MNR fine-tuning for the German dataset with randomly sampled negatives

The Figure 22 shows the four accuracy plots for the MNR fine-tuning on the German dataset with randomly sampled negatives, with the summary of the models used as a starting point shown in Table 18.

Color of the Curve	Model starting point for MNR training
Green	XLM-RoBERTa checkpoint, adapted on job-ads checkpoint and fine-tuned via MLM
Blue	XLM-RoBERTa checkpoint, adapted on job-ads checkpoint
Orange	XLM-RoBERTa out-of-the-box checkpoint, previously fine-tuned via MLM
Purple	XLM-RoBERTa out-of-the-box checkpoint

Table 18: Descriptions of accuracy plots in Figure 22 of random sampled negative triplet data for MNR fine-tuning for German dataset

The accuracy plots shown in Figure 22 reveal that the XLM-RoBERTa checkpoint that is not adapted for any specific task is the least accurate among the 4 variants tested in the MNR fine-tuning process. The remaining variants are capable of achieving an accuracy score close to 98% as the fine-tuning process continues. The domain-adapted XLM-RoBERTa models that are adapted to the job-ad data and further fine-tuned via MLM task perform the best, similar to the findings from the hard negative sampling experiments. It can also be seen that the accuracy in the initial stages for the XLM-RoBERTa model adapted on both the job-ads domain and the ESCO ontology domain has a higher accuracy when compared to the same model variant on English dataset as shown in Figure 20. This must be due to the face that this model checkpoint is adapted on the job-ads domain data where all the domain data was in the German language.

Based on the findings from the accuracy plots of the MNR fine-tuning task shown in Figure 19, 20, 21, and 22, the following conclusions can be drawn:

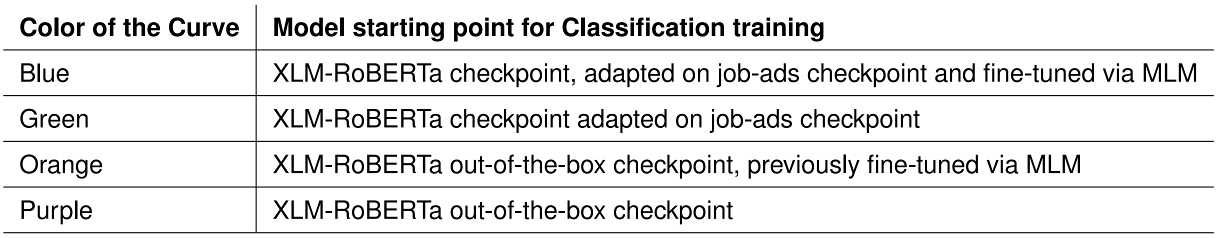
1. The performance of all MNR fine-tuning variants is lower when using the XLM-RoBERTa out-of-the-box checkpoint as a starting point.
2. Domain adaptation on job-ads data, along with initial fine-tuning via MLM on the same checkpoint, consistently yielded better results compared to checkpoints that only underwent MLM on the XLM-RoBERTa out-of-the-box model.

Based on the two points mentioned earlier, it can be inferred that adapting the model to the specific domain is crucial in achieving better performance on the MNR task and enhancing it.

6.1.3 Classification Task

In the classification task, we started with 4 different variants of models, including 2 fine-tuned checkpoints from the MLM task, the out-of-the-box XLM-RoBERTa model, and the XLM-RoBERTa checkpoint adapted to the job-ads domain. Each language had these 4 variants as starting points for training using the classification objective. We randomly selected 10% of the overall classification data pairs for validation in both the English and German datasets and the data had binary labels with '0' indicating a positive pair and '1' indicating a negative pair, as mentioned in Section 4.3.2 and Section 4.3.3.

The summary of the model starting points used in the Figure 23 is shown in the Table 19.



Color of the Curve	Model starting point for Classification training
Blue	XLM-RoBERTa checkpoint, adapted on job-ads checkpoint and fine-tuned via MLM
Green	XLM-RoBERTa checkpoint adapted on job-ads checkpoint
Orange	XLM-RoBERTa out-of-the-box checkpoint, previously fine-tuned via MLM
Purple	XLM-RoBERTa out-of-the-box checkpoint

Table 19: Descriptions of accuracy plots of classification task on the English datasets in Figure 23

From the above accuracy plots, we can that using the XLM-RoBERTa out-of-the-box checkpoint as the starting point for classification training does not yield a good

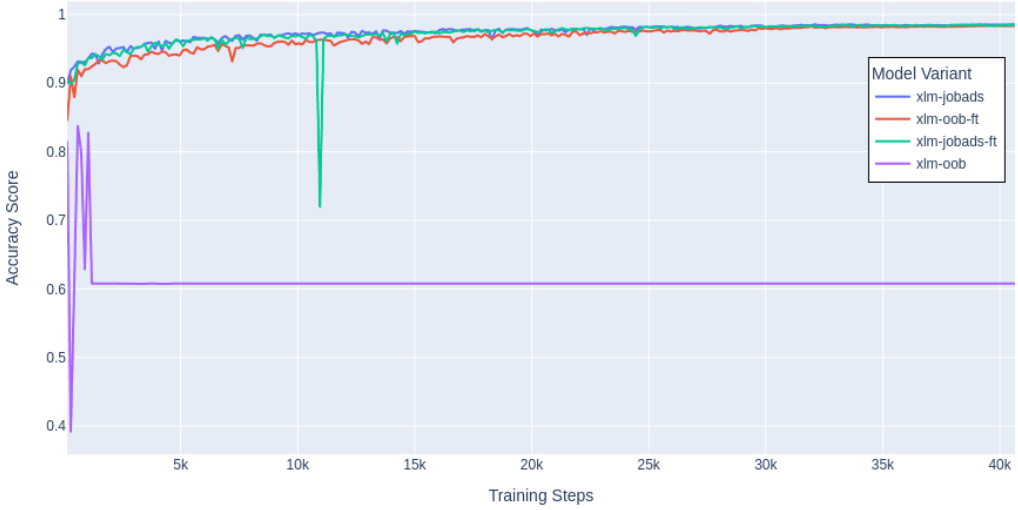


Figure 23: Plots of the accuracy scores of the Classification fine-tuning on the English dataset

accuracy score. It seems that this variant of the model is stuck on an accuracy of 60%, and is unable to improve even with further fine-tuning. On the other hand, all variants that had initial domain adaptation had an accuracy closer to 98%. These results imply that starting models with domain adaptation perform better in classifying positive and negative pairs from the classification data.

Figure 24 contains four accuracy plots for the Classification training on the English dataset, with the summary of the models used as a starting point shown in Table 20.

Color of the Curve	**Model starting point for Classification training**
Pink	XLM-RoBERTa checkpoint, adapted on job-ads checkpoint and fine-tuned via MLM
Grey	XLM-RoBERTa checkpoint, adapted on job-ads checkpoint
Orange	XLM-RoBERTa out-of-the-box checkpoint, previously fine-tuned via MLM
Red	XLM-RoBERTa out-of-the-box checkpoint

Table 20: Descriptions of accuracy plots of classification task on the German datasets in Figure 24

From the accuracy plots in Figure 24, we can that using the XLM-RoBERTa out-of-the-box checkpoint as the starting point for classification training does not yield a good accuracy score. It seems that this variant of the model is stuck on an accuracy

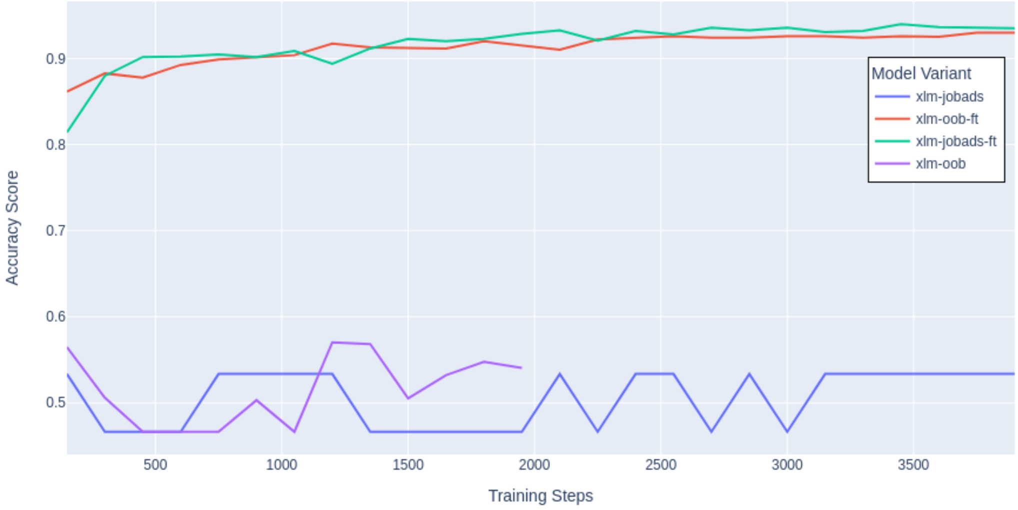


Figure 24: Plots of the accuracy scores of the Classification fine-tuning on the German dataset

of 60%, and is unable to improve even with further fine-tuning. This behavior is also observed for the XLM-RoBERTa checkpoint that is domain-adapted on the job-ads checkpoint. On the other hand, variants that had MLM fine-tuning done initially before training via the classification task had an overall accuracy closer to 98%. These results imply that starting models with domain adaptation perform better in classifying positive and negative pairs from the classification data.

From the results shown in Figure 23 and Figure 24, we can conclude the following points:

1. The classification task does not improve the overall accuracy of the models per se for our experiments, as we do not see a lot of improvement in the value of accuracy from the evaluation curves of the classification task.
2. Domain adaptation on the job-ads data, along with further fine-tuning via MLM on the same checkpoint yields better results compared to checkpoints that only underwent MLM on the XLM-RoBERTa out-of-the-box model.
3. Choosing an out-of-the-box model variant for the task of classification does not yield a higher accuracy when used as a starting point when compared to models that had either domain adaptation or fine-tuning or both applied to them

The poor performance of the classification task might be due to the default hyperparameter setting that is used for the classification task, and further experimentation with different hyperparameters was not conducted as the initial iterations with the classification task did not yield promising results. Instead, more time was invested in the MNR strategy to try and improve it as the models fine-tuned via the MNR task showed more promising results. However, it can still be seen that models that are domain adapted, either to the job-ads domain or the ESCO ontology domain, are the better-performing models when it comes to the task of classification in comparison to out-of-the-box models.

6.1.4 Conclusions from the Internal Evaluations

From all the observations we see in the Internal Evaluation section, we can summarize and make the following conclusions:

1. All model variants do not have a high accuracy in the initial stages of the fine-tuning tasks. However, we can see as the training continues, the accuracy scores of different fine-tuning tasks improve.
2. Models that are domain adapted, either on the job-ads data or the ESCO ontology data or both perform better over time in comparison to models that are not domain adapted.
3. The XLM-RoBERTa variant fine-tuned on the job-ads data performs better on the German language datasets in comparison to the English language datasets.
4. The models not fine-tuned via the MLM task do not perform well on the classification tasks even as training continues for our experiments. Moreover, the models from the classification task do not seem as promising as the models from the MNR task, but this might be due to the default hyperparameter values taken for the classification task.

6.2 External Evaluation of Models

From Section 6.1 on internal evaluation, we could see that the domain-adaptation tasks on different variants of models helped the model improve in performance on the validation dataset. However, the actual performance on the end task from each of these model variants might be different, and thus an external evaluation for each of these models is conducted.

6.2.0.1 Data for External Evaluation

As mentioned in Section 3.3, 2 gold standard datasets have been created to evaluate the end task. Each of the gold standard datasets contains job-ad terms, along with the terms surrounding the job-ad term, thus acting as a context for the central job-ad term. The two variants consist of a random sample of 25 job-ad terms that frequently appear in candidate applications and a challenge sample of 15 job-ad terms that contain some of the more tailored and challenging terms that are harder to classify correctly at human-level annotations. As mentioned in Section 3.3, this data is created in the Paper Gnehm et al. [2022a].

6.2.0.2 Evaluation Data Annotation

As the evaluation data was created in the Paper Gnehm et al. [2022a], we already have a dataset that is annotated by human annotators. In this dataset, we already have annotations for 268 classes from the random gold standard dataset and 228 classes from the challenge gold standard dataset. These class suggestions are from the model variants from the Paper Gnehm et al. [2022a], wherein the top suggestions for each model variant are collected and annotated. For the 15 model variants that we trained, we take the top-class suggestion made by each model variant for annotation via 3 human annotators, wherein the top-class suggestions are selected for both the English and German language models.

The class suggestions made by each model are one of the 638 lowest-level classes available in the ESCO ontology. The job-ad terms that best fall into any of the classes of the ESCO ontology are the top classes suggested by each of the model variants. As part of annotations, each annotator gives a score of 0, 0.5, and 1 for the top suggestions given by each of the models. The scores represent that the annotator found the suggestion not related at all, a little relevant to the input term, and fully relevant to the input term, respectively. A mean score from each of the annotator's scores is computed, and this mean score is used for assessing the performance of each model variant. For all three annotators, the inter-annotator agreement score is computed via Krippendorff's alpha[1] coefficient, which is a statistical measure of the agreement achieved when labeling the gold standard data. Krippendorff's alpha is computed for two different cases, once on the data that is only suggested by the 15 new model variants from this thesis, and once for the whole annotation dataset. The following Table 21 gives a summary of Krippendorff's alpha coefficient for the challenge and the random gold standard evaluations, on the data that is added by

[1] Krippendorff's alpha: `https://en.wikipedia.org/wiki/Krippendorff%27s_alpha`

the new model variants and the overall available annotation data.

Gold Standard Dataset	Krippendorff's alpha
Random Gold Standard (New Class Dataset)	0.861
Challenge Gold Standard (New Class Dataset)	0.847
Random Gold Standard (Whole Dataset)	0.835
Challenge Gold Standard (Whole Dataset)	0.714

Table 21: Krippendorff's alpha coefficients for the random and the challenge gold standard datasets. The new class dataset represents the new data that is added to the original annotation datasets, and the whole dataset represents the data with all class annotations combined.

A lower Krippendorff's alpha on the whole challenge gold standard dataset in contrast to the alpha value of the whole random gold standard dataset shows that the inter-annotator agreement for terms in the challenge gold standard dataset is lower, meaning that not all annotators are able to agree on common ground while assigning labels. This is due to the nature of the terms included in the challenge dataset, which are tougher at human-level annotations. We can also see that for the new class annotations, Krippendorff's alpha values are higher, implying that all the annotators agree more on the scores they give to the newly suggested classes. This is also because many of the new class suggestions provided by the models were not very relevant to the input job ad mention, and all annotators agreed on discarding the suggestions equally.

6.2.0.3 Metric for External Evaluation

The evaluation metric used for our study is Mean Average Precision (mAP), which is calculated at different hit levels. In simple terms, mAP at hits indicates the number of relevant documents present in the top results, where hits can be varied based on the number of available results. This metric is denoted as mAP@k, where k is the number of top results considered. Our study computes mAP@1 and mAP@5, which means we evaluate the mAP of the top result and the top 5 results returned by each model variant. The mAP is a useful metric for evaluating the quality of ranking algorithms and understanding their performance. It provides a single number that reflects the precision and recall of the ranking results. The mAP is calculated as per the equation shown below from Manning et al. [2008]:

$$mAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_{\mathrm{j}}} \sum_{k=1}^{m_{\mathrm{j}}} Precision\,(R_{\mathrm{jk}})$$

In this equation, Q is the number of input queries, which can be 25 for the random gold standard dataset and 15 for the challenge gold standard dataset. M is the total number of suggestions that are accepted, and K is the cutoff rank which defines how many of the top suggestions to consider for calculating the precision. Based on the formulation of the mAP, we compute mAP@1 and mAP@5 for our external evaluation. mAP@1 evaluation metric only considers the top term suggested by each model, and in theory, only computes the precision given by each model variant. On the other hand, the mAP@5 evaluation metric computes the mAP and the ranking of the top 5 suggestions made by each model variant and gives a better idea of how each model variant suggests the most 5 related values across all of the ESCO ontology. The mAP@5 evaluation metric thus helps better generalize the suggestions that each of the models provides in comparison to the mAP@1 evaluation metric.

For the cutoff rank, for each model annotation to be accepted as a true positive, we have two different thresholds that we use. We use both soft and hard precision boundaries for each mAP@1 and mAP@5 metric we compute. These boundaries determine whether a threshold has been reached for the mean annotation score to be considered a true positive. The soft precision boundary is set at 0.3, and the hard precision boundary is set at 0.6. This means that if the mean of the annotation scores for a particular class is greater than 0.3 and the soft precision boundary is used, then that prediction is considered a true positive. On the contrary, if there is a class in the top 5 suggestions that is not annotated by human annotators, then we consider that particular class to be a false positive and assign it a score of 0. Also, the top 5 suggestions made by each of the 15 model variants could be either from the original gold standard dataset or from the new top-class suggestions added for each of the gold standard datasets. In both cases, the mean of the annotation score is taken as a basis to decide whether the class suggestion is a true positive or a false positive.

Based on the above conditions on the evaluation metrics, the external evaluation on the top 20 classes suggested by each of the model variants is computed.

6.2.0.4 Naming Convention for Models in External Evaluations

To help the reader understand which model is represented in each row of the evaluation tables, a naming convention is followed. The convention uses '_' to represent different sections in the model variant name.

The first section of the model variant name indicates the fine-tuning method used, which can be 'mlm', 'mnr', or 'classification', as explained in Chapter 4. The second section, separated by the '_', represents the type of model checkpoint used as a starting point for the fine-tuning task. The second half of the model variant name may include either a single model variant or a model variant fine-tuned via MLM.

For 'mnr' model variants, the first half of the name also includes the variant of the data used, either 'hardneg' for hard negative sampled triplet datasets or 'randneg' for randomly sampled negative triplet datasets for the fine-tuning task.

Table 22 provides a brief explanation of all the components used to name the model variants in the external evaluation tables.

Model Name Component	**Representation of the component**
mnr-hardneg	MNR task with hard negative sampled triplet dataset
mnr-randneg	MNR task with randomly sampled negative triplet dataset
xlm-jobads	XLM-RoBERTa checkpoint domain adapted to the job-ads data
xlm-oob	XLM-RoBERTa out-of-the-box checkpoint
xlm-oob-ft-*x*	XLM-RoBERTa out-of-the-box checkpoint finetuned via MLM, where *x* represents the step of the checkpoint
xlm-jobads-ft-*x*	XLM-RoBERTa checkpoint domain adapted on the job-ads data and finetuned via MLM, where *x* represents the step of the checkpoint
classification	Classification task with a starting point mentioned in the other half of the model name

Table 22: Descriptions of components used in naming the model variants for external evaluation

6.2.1 Random Dataset Evaluations

This section covers the computation of the mAP@1 and mAP@5 evaluation metrics for the random gold standard dataset. As outlined in Section 3.3, the random gold standard dataset contains 25 examples of terms that appear in German job ads in Switzerland.

The subsections in Section 6.2.1 contain evaluation tables with columns labeled EDU, EXP, LNG, ALL, and R. These columns represent the mAP score for educational terms, experience terms, language terms, all terms combined, and the rank of each term section, respectively. The rank column is included with each term's mAP score to facilitate the comparison of model variants based on the type of term section being used.

6.2.1.1 mAP@1 Results for the External Evaluation

As an initial measure of evaluation, the mAP@1 evaluation metric was computed from the annotation scores. Table 23 gives the mAP scores and rankings of all 15 model variants. From this ranking, we can see that the MNR variant with the XLM-RoBERTa model adapted on the job-ads domain and further fine-tuned via MLM task is the best-performing model variant across the overall data. Except for the job-ad terms related to experience skills, this model variant is also the best-performing model across the job-ad term related to education skills and language skills. On the other hand, the worst-performing model across all the term sections and overall is the classification model with the XLM-RoBERTa out-of-the-box model used as a starting point. It can also be observed that the MNR variants with the hard negative sampling (*‘hardneg’*) data are better in suggesting the top term in comparison to MNR variants with randomly sampled negative (*‘randneg’*) data, with an overall increase of 20pp.

Model Variant	**EDU**	**R**	**EXP**	**R**	**LNG**	**R**	**ALL**	**R**
mnr-hardneg_xlm-jobads-ft-4650	0.600	1	0.600	6	1.000	1	0.680	1
mnr-randeng_xlm-jobads	0.200	10	0.700	1	1.000	1	0.560	2
xlm-jobads	0.400	2	0.700	1	0.400	6	0.520	3
mnr-randneg_xlm-jobads-ft-4650	0.300	3	0.700	1	0.400	6	0.480	4
mnr-hardneg_xlm-ft-320	0.300	3	0.700	1	0.400	6	0.480	4
mnr-randneg_xlm-oob	0.300	3	0.700	1	0.400	6	0.480	4
mnr-hardneg_xlm-oob	0.200	10	0.500	7	1.000	1	0.480	4
xlm-jobads-ft-4650	0.300	3	0.500	7	0.400	6	0.400	8
classification_xlm-jobads	0.100	13	0.400	9	0.800	4	0.360	9
classification_xlm-jobads-ft-4650	0.200	10	0.100	13	0.800	4	0.280	10
xlm-oob-ft-320	0.300	3	0.200	10	0.400	6	0.280	10
xlm-randneg_xlm-ft-320	0.300	3	0.200	10	0.400	6	0.280	10
mnr-randneg_xlm-oob	0.300	3	0.200	10	0.400	6	0.280	10
classification_xlm-oob-ft-320	0.000	14	0.000	14	0.400	6	0.080	14
classification_xlm-oob	0.000	14	0.000	14	0.000	15	0.000	15

Table 23: mAP@1 with the hard threshold for the English model variants on the random set

Table 24 shows the mAP@1 evaluation results for the random gold standard dataset for the German language. From this table, we can see the results are similar in

Model Variant	EDU	R	EXP	R	LNG	R	ALL	R
mnr-hardneg_xlm-jobads-ft-6450	0.800	1	0.600	6	1.000	1	0.760	1
mnr-randeng_xlm-jobads	0.300	4	0.700	1	1.000	1	0.600	2
xlm-jobads	0.400	2	0.700	1	0.600	6	0.560	3
xlm-jobads-ft-6450	0.400	2	0.600	6	0.600	6	0.520	4
mnr-hardneg_xlm-ft-600	0.300	4	0.700	1	0.400	8	0.480	5
mnr-randneg_xlm-oob	0.300	4	0.700	1	0.400	8	0.480	5
mnr-randneg_xlm-jobads-ft-6450	0.300	4	0.700	1	0.400	8	0.480	5
mnr-hardneg_xlm-oob	0.200	11	0.500	8	1.000	1	0.480	5
classification_xlm-jobads	0.100	13	0.400	9	1.000	1	0.400	9
classification_xlm-jobads-ft-6450	0.200	11	0.300	10	0.800	5	0.360	10
xlm-randneg_xlm-ft-600	0.300	4	0.300	10	0.400	8	0.320	11
mnr-randneg_xlm-oob	0.300	4	0.300	10	0.400	8	0.320	11
xlm-oob-ft-600	0.300	4	0.300	10	0.400	8	0.320	11
classification_xlm-oob-ft-600	0.000	14	0.000	14	0.400	8	0.080	14
classification_xlm-oob	0.000	14	0.000	14	0.000	15	0.000	15

Table 24: mAP@1 with the hard threshold for the German model variants on the random set

ranking to the results we see in Table 23, where the best performing model over all terms is the XLM-RoBERTa model with domain adaptation on job-ads data and further fine-tuned via MLM on the ESCO ontology data. On the other hand, the worst performing model is again the classification model with XLM-RoBERTa out-of-the-box model used as a starting point. It can also be seen that the mAP scores of the models for the German language are in general higher in comparision to models for the English language for XLM-RoBERTa checkpoint adapted to the job-ads data. This is because the job-ads data was in the German language, and thus the models for the German language perform better on the same random sample dataset.

It's important to note that these results are based on the hard threshold for true positives, and it is beneficial to also evaluate the performance of these models with soft thresholds and higher hit rates for the mAP evaluation metric.

The Table 25 and 26 represent the mAP@1 evaluation metric results for the random gold standard dataset with the soft threshold. The results seen in both these tables

Model Variant	EDU	R	EXP	R	LNG	R	ALL	R
mnr-hardneg_xlm-jobads-ft-4650	0.600	1	0.700	1	1.000	1	0.720	1
mnr-randeng_xlm-jobads	0.300	3	0.700	1	1.000	1	0.600	2
xlm-jobads	0.400	2	0.700	1	0.400	6	0.520	3
mnr-randneg_xlm-jobads-ft-4650	0.300	3	0.700	1	0.400	6	0.480	4
mnr-hardneg_xlm-ft-320	0.300	3	0.700	1	0.400	6	0.480	4
mnr-randneg_xlm-oob	0.300	3	0.700	1	0.400	6	0.480	4
mnr-hardneg_xlm-oob	0.200	11	0.500	7	1.000	1	0.480	4
xlm-jobads-ft-4650	0.300	3	0.500	7	0.400	6	0.400	8
classification_xlm-jobads	0.100	13	0.400	9	0.800	4	0.360	9
xlm-oob-ft-320	0.300	3	0.300	10	0.400	6	0.320	10
xlm-randneg_xlm-ft-320	0.300	3	0.300	10	0.400	6	0.320	10
mnr-randneg_xlm-oob	0.300	3	0.300	10	0.400	6	0.320	10
classification_xlm-jobads-ft-4650	0.200	11	0.100	13	0.800	4	0.280	13
classification_xlm-oob-ft-320	0.000	14	0.100	13	0.400	6	0.120	14
classification_xlm-oob	0.000	14	0.000	15	0.000	15	0.000	15

Table 25: mAP@1 with the soft threshold for the English model variants on the random set

are similar to the results seen in Tables 23 and 24. The MNR model variant with the XLM-RoBERTa adapted on the job-ads data and the ESCO ontology data is the best performing model variant overall. Because the soft threshold is used, we can see that the overall mAP values are higher for the results of both languages in comparison to the results with hard threshold.

Based on the results we see in Tables 23, 24, 25, and 26, we can make the following conclusions:

1. The best-performing model in all the cases is the MNR variant adapted on the job-ads data and further fine-tuned via the MLM task on the ESCO ontology data. This would imply that domain adaptation is beneficial for the end task of classifying job-ad terms onto the ESCO ontology.
2. The MNR variants with the hard negative sampled data are better in performance when compared to the same model variants with randomly sampled negative data, thus implying that the hard negative sampling does yield better

Model Variant	EDU	R	EXP	R	LNG	R	ALL	R
mnr-hardneg_xlm-jobads-ft-6450	0.800	1	0.700	1	1.000	1	0.800	1
mnr-randeng_xlm-jobads	0.400	2	0.700	1	1.000	1	0.640	2
xlm-jobads	0.400	2	0.700	1	0.600	6	0.560	3
xlm-jobads-ft-6450	0.400	2	0.600	7	0.600	6	0.520	4
mnr-hardneg_xlm-oob	0.300	8	0.500	8	1.000	1	0.520	4
mnr-hardneg_xlm-ft-600	0.300	8	0.700	1	0.400	11	0.480	6
xlm-randneg_xlm-ft-600	0.400	2	0.500	8	0.600	6	0.480	6
mnr-randneg_xlm-oob	0.300	8	0.700	1	0.400	11	0.480	6
mnr-randneg_xlm-jobads-ft-6450	0.300	8	0.700	1	0.400	11	0.480	6
mnr-randneg_xlm-oob	0.400	2	0.500	8	0.600	6	0.480	6
xlm-oob-ft-600	0.400	2	0.500	8	0.600	6	0.480	6
classification_xlm-jobads	0.200	12	0.400	12	1.000	1	0.440	12
classification_xlm-jobads-ft-6450	0.200	12	0.300	13	0.800	5	0.360	13
classification_xlm-oob-ft-600	0.000	14	0.100	14	0.400	11	0.120	14
classification_xlm-oob	0.000	14	0.000	15	0.000	15	0.000	15

Table 26: mAP@1 with the soft threshold for the German model variants on the random set

results on the end task for the models.

3. The classification task with the artificially generated labels is the worst-performing model in all the results. This would imply that a strategy of artificially labeling the data is not the best for domain adaptation and did not work for our experiments.

4. Changing the threshold for the true positives to either a harder or a softer threshold does not make a difference in the best and the worst performing variant of the model.

Since the random gold standard dataset contains terms that are chosen at random from job applications, the results we see here can be generalized across other terms that might appear in the job markets in Switzerland. We can now look at the evaluation metrics for the random gold standard dataset with the mAP@5 evaluation metric to see how well the model variants perform on those terms.

6.2.1.2 mAP@5 Results for the External Evaluations

Model Variant	EDU	R	EXP	R	LNG	R	ALL	R
mnr-hardneg_xlm-jobads-ft-4650	0.654	1	0.621	6	0.961	1	0.702	1
mnr-randeng_xlm-jobads	0.460	2	0.700	4	0.940	3	0.652	2
xlm-jobads	0.426	4	0.668	5	0.592	7	0.556	3
xlm-jobads-ft-4650	0.437	3	0.605	7	0.641	6	0.545	4
mnr-hardneg_xlm-oob	0.292	6	0.557	8	0.961	1	0.532	5
mnr-randneg_xlm-jobads-ft-4650	0.266	10	0.723	1	0.556	9	0.507	6
mnr-hardneg_xlm-ft-320	0.266	10	0.723	1	0.556	9	0.507	6
mnr-randneg_xlm-oob	0.266	10	0.723	1	0.556	9	0.507	6
classification_xlm-jobads	0.087	14	0.520	9	0.800	4	0.403	9
classification_xlm-jobads-ft-4650	0.330	5	0.248	11	0.800	4	0.391	10
xlm-oob-ft-320	0.290	7	0.248	11	0.351	12	0.286	11
xlm-randneg_xlm-ft-320	0.290	7	0.248	11	0.351	12	0.286	11
mnr-randneg_xlm-oob	0.290	7	0.248	11	0.351	12	0.286	11
classification_xlm-oob-ft-320	0.103	13	0.250	10	0.590	8	0.259	14
classification_xlm-oob	0.000	15	0.000	15	0.000	15	0.000	15

Table 27: mAP@5 with the hard threshold for the English model variants on the random set

Table 27 presents the results of the mAP@5 evaluation metric on the English language models for the random gold standard dataset. Similarly, Table 28 displays the same metric's results on the German language models. Based on the tables, we observe that the MNR model with the XLM-RoBERTa model adapted on job-ads data and further fine-tuned on the ESCO ontology data via MLM as a starting point remains the best-performing model, as previously seen in Section 6.2.1. Conversely, the classification model with the XLM-RoBERTa out-of-the-box model used as a starting point performs the worst for mAP@5 evaluation metric results with a hard threshold. As seen in earlier results, MNR variants with hard negative sampled triplet data outperform the same model variant with randomly sampled negative data.

Table 29 and 30 show the results of the mAP@5 evaluation metric with the soft threshold for the English and the German language models on the random gold standard dataset. The same observations as seen in Tables 27 and 28 can be drawn in these results for the best and the worst-performing models overall.

Model Variant	EDU	R	EXP	R	LNG	R	ALL	R
mnr-hardneg_xlm-jobads-ft-6450	0.763	1	0.621	7	1.000	1	0.754	1
mnr-randeng_xlm-jobads	0.605	2	0.700	4	0.973	3	0.717	2
xlm-jobads-ft-6450	0.498	3	0.652	6	0.736	7	0.607	3
xlm-jobads	0.406	4	0.668	5	0.771	6	0.584	4
mnr-hardneg_xlm-ft-600	0.324	6	0.723	1	0.623	8	0.543	5
mnr-randneg_xlm-oob	0.324	6	0.723	1	0.623	8	0.543	5
mnr-randneg_xlm-jobads-ft-6450	0.324	6	0.723	1	0.623	8	0.543	5
mnr-hardneg_xlm-oob	0.292	9	0.557	8	1.000	1	0.540	8
classification_xlm-jobads-ft-6450	0.330	5	0.398	10	0.800	5	0.451	9
classification_xlm-jobads	0.107	13	0.520	9	0.950	4	0.441	10
xlm-randneg_xlm-ft-600	0.290	10	0.348	11	0.390	12	0.333	11
mnr-randneg_xlm-oob	0.290	10	0.348	11	0.390	12	0.333	11
xlm-oob-ft-600	0.290	10	0.348	11	0.390	12	0.333	11
classification_xlm-oob-ft-600	0.103	14	0.250	14	0.590	11	0.259	14
classification_xlm-oob	0.050	15	0.000	15	0.000	15	0.020	15

Table 28: mAP@5 with the hard threshold for the German model variants on the random set

For all the results with the mAP@5 evaluation metric on the random gold standard dataset, the average percentage of overall classes that had no mean annotation score linked to them was 26.32%. In other words, on average, for all the top 5 suggestions made by each model variant, there was a class that had no mean annotation score and was thus considered to be a false negative class with a mean annotation score of 0. This would imply that the mAP scores could be higher overall if all the top 5 suggested classes for each term were annotated, as they would not have been considered false negatives by default. However, despite there being classes that have no annotations for them, we can see no changes in the overall results we see for the mAP@1 and mAP@5 evaluation metrics for the English and German language models.

From the results and observations seen in Tables 27, 28, 29, 30 for the model evaluations on the random gold standard dataset, we can make the following conclusions:

1. The MNR variant with the XLM-RoBERTa model adapted both on the job-ads

Model Variant	EDU	R	EXP	R	LNG	R	ALL	R
mnr-hardneg_xlm-jobads-ft-4650	0.657	1	0.699	5	0.961	1	0.734	1
mnr-randeng_xlm-jobads	0.496	2	0.776	1	0.940	3	0.697	2
xlm-jobads	0.301	6	0.615	8	0.961	1	0.559	3
xlm-jobads-ft-4650	0.426	4	0.668	6	0.592	7	0.556	4
mnr-hardneg_xlm-oob	0.437	3	0.629	7	0.641	6	0.555	5
mnr-randneg_xlm-jobads-ft-4650	0.289	10	0.765	2	0.556	9	0.533	6
mnr-hardneg_xlm-ft-320	0.289	10	0.765	2	0.556	9	0.533	6
mnr-randneg_xlm-oob	0.289	10	0.765	2	0.556	9	0.533	6
classification_xlm-jobads	0.330	5	0.339	14	0.800	4	0.428	9
classification_xlm-jobads-ft-4650	0.087	14	0.538	9	0.800	4	0.410	10
xlm-oob-ft-320	0.290	7	0.363	10	0.351	12	0.332	11
xlm-randneg_xlm-ft-320	0.290	7	0.363	10	0.351	12	0.332	11
mnr-randneg_xlm-oob	0.290	7	0.363	10	0.351	12	0.332	11
classification_xlm-oob-ft-320	0.107	13	0.350	13	0.590	8	0.301	14
classification_xlm-oob	0.000	15	0.000	15	0.000	15	0.000	15

Table 29: mAP@5 with the soft threshold for the English model variants on the random set

data and the ESCO ontology data is the best-performing model in all the cases, implying that domain adaptation is helpful for the end task of classification of job-ad terms onto the ontology.

2. The MNR variant with the hard negative sampled data has better mAP when compared to the same variants with randomly sampled negatives, with an average increase of 31pp overall evaluation metrics. This implies that hard negative sampling does have a positive impact on the final suggestions made by model variants.

3. The MNR variants is the best-performing model on mAP@1 and mAP@5 evaluation metric, implying that MNR helps in generating class suggestions that are useful overall instead of just increasing the precision of the best-suggested class.

4. The classification task with artificially generated labels for the input terms is always the worst-performing model, implying that using artificial labels for

Model Variant	EDU	R	EXP	R	LNG	R	ALL	R
mnr-hardneg_xlm-jobads-ft-6450	0.765	1	0.699	5	1.000	1	0.785	1
mnr-randeng_xlm-jobads	0.641	2	0.776	1	0.973	3	0.762	2
xlm-jobads-ft-6450	0.503	3	0.676	6	0.736	7	0.619	3
xlm-jobads	0.425	4	0.668	7	0.771	6	0.592	4
mnr-hardneg_xlm-ft-600	0.333	12	0.615	8	1.000	1	0.579	5
mnr-randneg_xlm-oob	0.348	8	0.765	2	0.623	8	0.570	6
mnr-randneg_xlm-jobads-ft-6450	0.348	8	0.765	2	0.623	8	0.570	6
mnr-hardneg_xlm-oob	0.348	8	0.765	2	0.623	8	0.570	6
classification_xlm-jobads-ft-6450	0.390	5	0.563	9	0.590	11	0.499	9
classification_xlm-jobads	0.390	5	0.563	9	0.590	11	0.499	9
xlm-randneg_xlm-ft-600	0.390	5	0.563	9	0.590	11	0.499	9
mnr-randneg_xlm-oob	0.207	13	0.538	12	0.950	4	0.488	12
xlm-oob-ft-600	0.339	11	0.447	13	0.800	5	0.474	13
classification_xlm-oob-ft-600	0.107	14	0.350	14	0.590	11	0.301	14
classification_xlm-oob	0.050	15	0.000	15	0.000	15	0.020	15

Table 30: mAP@5 with the soft threshold for the German model variants on the random set

the input data does not yield good results for our experiments. This might be due to the default hyperparameter settings that are used for the task and not completely because of the artificial labeling strategy.

5. The mAP scores for the XLM-RoBERTa model adapted on both the job-ads data and the ESCO ontology data are higher for the German language models in comparison to the English language models. This is because the job-ads data was only in the German language, and thus the German language models benefit from this fine-tuning more than the English language models.

6.2.2 Challenge Dataset Evaluations

This section covers the computation of the mAP@1 and mAP@5 evaluation metrics for the challenge gold standard dataset. As outlined in Section 3.3, the challenge gold standard dataset contains 15 tailored examples of terms that appear in job ads that are harder to classify at the human level, and also specific job-ad terms that appear more frequently only in job markets and applications in Switzerland.

The subsections in Section 6.2.2 contain evaluation tables with columns labeled mAP@x-y and Rank. The x term in the 'mAP' column of the evaluation tables can be either '1' or '5' based on either the mAP@1 evaluation metric or the mAP@5 evaluation metric, respectively. The y term in the evaluation tables can be either 'hard' or 'soft' depending on which threshold is used for computing the true positives from the mean annotation scores.

6.2.2.1 mAP@1 Results for the External Evaluation

To evaluate the performance of the top class suggested by each model variant for a given job-ad term, the initial step is to compute mAP@1 from the annotation scores. Table 31 displays the results of mAP@1 using the hard threshold boundary for the English dataset. Of all the four MNR variants with the hard negative sampled dataset, which includes using the out-of-the-box XLM-RoBERTa model, domain-adapted XLM-RoBERTa model on the job-ads dataset, and both models fine-tuned via MLM task. These results are consistent with the random gold standard dataset results discussed in Section 6.2.1, where MNR models were found to be the best-performing models for the English language using the hard negative sampled dataset. On the other hand, the out-of-the-box classification model variant performs the worst out of all 15 model variants for mAP@1 using the hard threshold on the English dataset. These results also align with the results seen in Section **??** on the random gold standard datasets, where the out-of-the-box classification model was found to be the worst-performing model variant.

Additionally, it is interesting to note that the MNR models with the hard negative sampled dataset outperform the MNR models with the randomly sampled negative triplet dataset, indicating the importance of using a high-quality training dataset for model fine-tuning. Furthermore, it is worth mentioning that the domain-adapted XLM-RoBERTa model performs slightly better than the out-of-the-box XLM-RoBERTa model, suggesting that fine-tuning a pre-trained language model on a domain-specific dataset can improve its performance on that particular domain. Overall, the results of the mAP@1 evaluation suggest that the MNR models with the hard negative sampled dataset, especially those fine-tuned via the MLM task, are the best-performing models for the English language job-ad classification task.

From Table 32, we can see that the results for the German dataset are consistent

Model Variant	mAP@1-hard	Rank
mnr-hardneg_xlm-jobads-ft-4650	0.733	1
mnr-randeng_xlm-jobads	0.533	2
mnr-randneg_xlm-jobads-ft-4650	0.400	3
mnr-hardneg_xlm-ft-320	0.400	3
mnr-randneg_xlm-oob	0.400	3
xlm-jobads	0.400	3
mnr-hardneg_xlm-oob	0.400	3
xlm-jobads-ft-4650	0.333	8
classification_xlm-jobads	0.333	8
classification_xlm-jobads-ft-4650	0.267	10
xlm-oob-ft-320	0.267	10
xlm-randneg_xlm-ft-320	0.267	10
mnr-randneg_xlm-oob	0.267	10
classification_xlm-oob-ft-320	0.067	14
classification_xlm-oob	0.067	14

Table 31: mAP@1 with the hard threshold for the English model variants on the challenge set

with the English dataset. The MNR variants with the hard negative sampled dataset are once again the best-performing models, and the XLM-RoBERTa models adapted to the job-ads domain and fine-tuned via MLM task are the top-performing models. It's interesting to note that the worst-performing model for the German dataset is also the classification model, which is consistent with the results for the English dataset. It can also be observed that the checkpoints adapted on the job-ads data and the ESCO ontology data have a higher mAP value for the German language models in comparison to the English language models.

Additionally, it can also be seen that the MNR variant with the hard negative sampled triplet dataset performs better in suggesting the top class when compared to the MNR variant with a randomly sampled negative triplet dataset. These results, along with the results seen in Table 31, would suggest that creating a triplet dataset that has hard sampled negatives is better in performance on the end task rather than randomly assigning negatives to an input term.

It's important to note that these results are based on the hard threshold for true positives, and it is be beneficial to also evaluate the performance of these models

Model Variant	mAP@1-hard	Rank
mnr-hardneg_xlm-jobads-ft-6450	0.778	1
mnr-randeng_xlm-jobads	0.667	2
mnr-hardneg_xlm-ft-600	0.556	3
xlm-jobads-ft-6450	0.556	3
mnr-randneg_xlm-oob	0.556	3
mnr-randneg_xlm-jobads-ft-6450	0.556	3
xlm-jobads	0.556	3
classification_xlm-jobads	0.556	3
classification_xlm-jobads-ft-6450	0.444	9
mnr-hardneg_xlm-oob	0.444	9
xlm-randneg_xlm-ft-600	0.111	11
mnr-randneg_xlm-oob	0.111	11
xlm-oob-ft-600	0.111	11
classification_xlm-oob	0.000	14
classification_xlm-oob-ft-600	0.000	14

Table 32: mAP@1 with the hard threshold for the German model variants on the challenge set

with soft thresholds and higher hit rates for the mAP evaluation metric.

Tables 33 and 34 display the results of the mean average precision at one (mAP@1) using a soft threshold boundary for the English and German dataset, respectively. The soft threshold considers a lower mean value of annotation scores as a true positive. As a result, the mAP values for model variants using the soft threshold in Table 33 are higher than those using the hard threshold in Table 31. The MNR variant with a hard negative sampled dataset for the English language performs the best among all 15 model variants, as shown in Table 33. The XLM-RoBERTa model, which is adapted to job advertisement data and fine-tuned using MLM, has the highest mAP among all 15 models. Also, the MNR model variants with a hard negative sampling dataset perform with an average of 30pp better in comparison to the same model variants with a random negative sampling dataset

From all the results seen in Tables 31, 32, 33, 34, we can make the following conclusions:

1. The MNR variant of the XLM-RoBERTa model adapted to the job-ads data and further fine-tuned via the MLM task on the ESCO ontology data with

Model Variant	mAP@1-soft	Rank
mnr-hardneg_xlm-jobads-ft-4650	0.733	1
xlm-jobads-ft-4650	0.667	2
mnr-randneg_xlm-jobads-ft-4650	0.600	3
mnr-hardneg_xlm-ft-320	0.600	3
mnr-randneg_xlm-oob	0.600	3
mnr-randeng_xlm-jobads	0.600	3
xlm-jobads	0.600	3
mnr-hardneg_xlm-oob	0.533	8
classification_xlm-jobads-ft-4650	0.467	9
xlm-oob-ft-320	0.400	10
xlm-randneg_xlm-ft-320	0.400	10
classification_xlm-jobads	0.400	10
mnr-randneg_xlm-oob	0.400	10
classification_xlm-oob-ft-320	0.133	14
classification_xlm-oob	0.133	14

Table 33: mAP@1 with the soft threshold for the English model variants on the challenge set

the hard negative sampled dataset is the best-performing model. This would imply that domain adaptation is beneficial for the end task of classification of the job-ad terms on the ESCO ontology classes.

2. The MNR variants with the hard negative sampling are better in performance than the variants with randomly sampled negatives, with an average of 29pp increase for the former over the latter. This implies that creating a hard-sampled triplet dataset yields in better precision scores on the end task.

3. The mAP values for the MNR variants adapted to the job-ads data have higher mAP in general for the German language models compared to the English language models.

4. The classification variants of the model with the out-of-the-box XLM-RoBERTa model or the XLM-RoBERTa model fine-tuned via the MLM task are the worst-performing model variants out of all the 15 models for our experiments.

5. The hard and the soft thresholds for the true positives do not change the best and the worst performing model variants for the mAP@1 evaluation metric.

Model Variant	mAP@1-soft	Rank
mnr-hardneg_xlm-jobads-ft-6450	0.889	1
xlm-jobads-ft-6450	0.778	2
mnr-randeng_xlm-jobads	0.667	3
mnr-hardneg_xlm-ft-600	0.556	4
mnr-randneg_xlm-oob	0.556	4
mnr-randneg_xlm-jobads-ft-6450	0.556	4
xlm-jobads	0.556	4
mnr-hardneg_xlm-oob	0.556	4
classification_xlm-jobads	0.556	4
classification_xlm-jobads-ft-6450	0.444	10
xlm-randneg_xlm-ft-600	0.222	11
mnr-randneg_xlm-oob	0.222	11
xlm-oob-ft-600	0.222	11
classification_xlm-oob	0.000	14
classification_xlm-oob-ft-600	0.000	14

Table 34: mAP@1 with the soft threshold for the German model variants on the challenge set

Thus, we can conclude from the above observations that domain adaptation on both the job-ads data and the ESCO ontology data is beneficial for the task of classifying the job-ad terms into the ESCO ontology classes. Also, the models fine-tuned via the MNR loss are the best-performing models, and the models fine-tuned via the classification task are the worst-performing models. Also, model variants adapted to the job-ads data have higher mAP for the German language models in comparison to the English language models.

6.2.2.2 mAP@5 Results for the External Evaluation

For the mAP@5 evaluation metric, we consider the top 5 class suggestions made by each of the model variants. This metric thus helps better generalize the suggestions made by each class by selecting the top 5 suggestions instead of the previously used mAP@1 evaluation metric, wherein we only consider the top class suggestion made by each model variant. In cases where a particular class of the top 5 suggestions is not present in the annotated data, then that particular class is assumed to be a false negative and assigned a mean score of 0 by default.

Model Variant	mAP@5-hard	Rank
mnr-hardneg_xlm-jobads-ft-4650	0.726	1
mnr-randeng_xlm-jobads	0.524	2
mnr-hardneg_xlm-oob	0.501	3
xlm-jobads	0.476	4
mnr-hardneg_xlm-ft-320	0.444	5
mnr-randneg_xlm-oob	0.444	5
mnr-randneg_xlm-jobads-ft-4650	0.387	7
xlm-jobads-ft-4650	0.385	8
classification_xlm-jobads	0.360	9
xlm-oob-ft-320	0.299	10
xlm-randneg_xlm-ft-320	0.299	10
mnr-randneg_xlm-oob	0.299	10
classification_xlm-jobads-ft-4650	0.248	13
classification_xlm-oob-ft-320	0.156	14
classification_xlm-oob	0.056	15

Table 35: mAP@5 with the hard threshold for the English model variants on the challenge set

Tables 35 and 36 represent the results of mAP@5 with the hard threshold for the English language and the German language model variants, respectively on the challenge set. The results seen in these tables are similar to the results seen for the mAP@1 evaluation metric from Section 6.2.1.2. It can also be seen that the mAP values for models that are adapted to the job-ads data are higher for the German language when compared to the models in the English language.

The Tables 37 and 38 represent the results of the English and the German model variants with the soft threshold for the mAP@5 evaluation metric on the challenge set, respectively. These results follow the same best and the worst-performing models as seen in Tables 35 and 36. The same conclusions can be drawn from the results seen in these two tables as well.

For all the results with the mAP@5 evaluation metric on the challenge gold standard dataset, the average percentage of overall classes that had no mean annotation score linked to them was 23.75%. In other words, on average, for all the top 5 suggestions made by each model variant, there was a class that had no mean annotation score and was thus considered to be a false negative class with a mean annotation score

Model Variant	mAP@5-hard	Rank
mnr-hardneg_xlm-jobads-ft-6450	0.818	1
mnr-randeng_xlm-jobads	0.652	2
mnr-hardneg_xlm-ft-600	0.639	3
mnr-randneg_xlm-oob	0.639	3
mnr-hardneg_xlm-oob	0.613	5
xlm-jobads	0.607	6
xlm-jobads-ft-6450	0.545	7
classification_xlm-jobads	0.540	8
mnr-randneg_xlm-jobads-ft-6450	0.534	9
classification_xlm-jobads-ft-6450	0.413	10
xlm-randneg_xlm-ft-600	0.240	11
mnr-randneg_xlm-oob	0.240	11
xlm-oob-ft-600	0.240	11
classification_xlm-oob-ft-600	0.120	14
classification_xlm-oob	0.000	15

Table 36: mAP@5 with the hard threshold for the German model variants on the challenge set

of 0. This would imply that the mAP values could have been higher than the values we have now. However, despite these missing class annotations, the best and the worst-performing models remain the same.

From the results seen in Tables 35, 36, 37, 38, we can conclude the following:

1. The MNR variant of the XLM-RoBERTa model adapted to the job-ads data and further fine-tuned via the MLM task on the ESCO ontology data with the hard negative sampled dataset is the best-performing model, thus implying that domain adaptation is beneficial in the classification of job-ad terms on the ESCO ontology classes.
2. The MNR variants with hard negative sampling consistently outperform the MNR variants with random negative sampling, thus implying that hard negative sampling is a better strategy for fine-tuning via MNR.
3. Models that use job-ads data have higher mAP for the German language in comparison to the German language since the job-ads data is in the German language.

Model Variant	mAP@5-soft	Rank
mnr-hardneg_xlm-jobads-ft-4650	0.784	1
mnr-hardneg_xlm-ft-320	0.656	2
mnr-randneg_xlm-oob	0.656	2
mnr-randeng_xlm-jobads	0.630	4
mnr-hardneg_xlm-oob	0.618	5
xlm-jobads	0.601	6
mnr-randneg_xlm-jobads-ft-4650	0.592	7
xlm-jobads-ft-4650	0.590	8
xlm-oob-ft-320	0.517	9
xlm-randneg_xlm-ft-320	0.517	9
mnr-randneg_xlm-oob	0.517	9
classification_xlm-jobads	0.468	12
classification_xlm-jobads-ft-4650	0.441	13
classification_xlm-oob-ft-320	0.222	14
classification_xlm-oob	0.144	15

Table 37: mAP@5 with the soft threshold for the English model variants on the challenge set

4. The classification variant of the model with the out-of-the-box XLM-RoBERTa model is the worst-performing model variant out of all the 15 models, implying that either data that has artificial labels for input data is not beneficial for the task of domain adaptation, or the default hyperparameter values for the classification task are not optimal for our experiments.

5. The hard and the soft thresholds for the true positives do not change the best and the worst performing model variants for the mAP@5 evaluation metric.

6.2.3 External Evaluation for Terms Without Context

For the external evaluations done without taking into account the surrounding words, i.e., context words for the input job-ad term, the class suggestions by each model variant were initially eyeballed to see if they are relevant. However, these class suggestions which only took into account the job-ad term did not seem to be related to the input job-ad term at all. They seemed to be almost random for all the job-ad terms, across the random and the challenge gold standard datasets. Since the

Model Variant	mAP@5-soft	Rank
mnr-hardneg_xlm-jobads-ft-6450	0.862	1
mnr-hardneg_xlm-oob	0.694	2
mnr-randeng_xlm-jobads	0.680	3
xlm-jobads-ft-6450	0.662	4
mnr-hardneg_xlm-ft-600	0.639	5
mnr-randneg_xlm-oob	0.639	5
xlm-jobads	0.625	7
classification_xlm-jobads	0.615	8
mnr-randneg_xlm-jobads-ft-6450	0.534	9
classification_xlm-jobads-ft-6450	0.451	10
xlm-randneg_xlm-ft-600	0.427	11
mnr-randneg_xlm-oob	0.427	11
xlm-oob-ft-600	0.427	11
classification_xlm-oob-ft-600	0.120	14
classification_xlm-oob	0.037	15

Table 38: mAP@5 with the soft threshold for the German model variants on the challenge set

results of models without using context words were not related to the input job-ad term, we decided to not annotate the class suggestions for model variants without using the surrounding context words.

6.2.4 Conclusions from the External Evaluations

From all the results seen in Section 6.2 on the random and the challenge gold standard, we can draw the following conclusions:

1. The best-performing model was the MNR variant adapted to both the job-ads data and the ESCO ontology data. This implies that domain adaptation on both the input and the target domain is beneficial for the LLMs to make the end task of mapping from the input domain onto the target domain.

2. The MNR variants with hard negative triplet data consistently performed better when compared to the same model variants with randomly sampled negative data. This implies that using the hard negative strategy for mining negative terms for the anchor term is beneficial to the end task of term mapping

onto the ontology.

3. It can also be seen that the mAP scores for the German language models are higher when compared to the mAP scores for the English language models for the MNR variants adapted to the job-ads domain. This is probably due to the fact that the job-ads data consisted only of terms in the German language, and thus the German language models benefit from them, giving them higher mAP values.

4. The classification task with the artificially created labels for sentence pairs seems to be the worst-performing model consistently. This could either be because of the data shuffling strategy that is used to create the data for the classification task, or the default hyperparameters that are used to fine-tune models via this task.

5. The initial eyeballing of class suggestions by the models that did not use the context words surrounding the job-ad terms, we found that the suggestions provided are off across all model variants. This would imply that providing surrounding context words along with the job-ad term computes rich word embeddings in contrast to only using the central job-ad term across all model variants.

7 Conclusions and Future Work

In this thesis, we explore methods for domain adaptation that help with the end task of classifying mentions from an input domain to concepts that are in a target ontology. There are many areas and domains where such mappings could be beneficial, including healthcare, general information, customer support, etc.

In our case, the end task to solve was to map job-ad skill requirement mentions from German-speaking job ads in Switzerland. For this task, 3 methods were mainly used: the Masked Language Modelling method, the Multiple Negative Ranking loss method, and the classification method for pre-training and fine-tuning. We can also look to use the models fine-tuned via one of these methods as a base for fine-tuning different models based on the three tasks.

For this thesis, the following research questions were outlined as per Section 1.1:

1. Which pre-training and fine-tuning methods such as Masked Language Modelling (MLM), Multiple Negative Ranking (MNR), and classification is helpful for domain adaptation and the end-task of mapping terms onto an ontology?
2. Does hard negative sampling yield better results than using random negative sampling for domain adaptation using the Multiple Negative Ranking loss?
3. Does a sentence classification approach on artificially created labels from sentence relationships prove to be beneficial for the domain adaptation of an LLM?
4. Would domain adaptation be necessary or improve the target task of mapping terms onto an ontology?

Based on the results of the internal and external evaluations of 15 model variants conducted in this thesis, we can answer the above research questions as follows:

1. The models fine-tuned via the MLM method on a particular domain is beneficial to be used as a starting point for different tasks for domain adaptation when compared to using the default, out-of-the-box LLMs as a starting point. Moreover, based on the observations made, the MNR task is the best-

performing model for the end task of mapping terms onto the ontology.

2. For MNR, using hard negative triplet sampling data always yields better results for the end task of classification when compared to the same MNR model variants with randomly sampled negative triplet data, with the former method yielding an average increase of 30pp[1] over the latter.

3. The method of pairing ontology terms and assigning them an artificial label is not beneficial for domain adaptation when compared to the MNR task of domain adaptation. However, it is also possible that the classification setting used for our experiments was not optimal in either the data created or the hyperparameters used.

4. Domain adaptation, on both the input texts and the target domain, is necessary to be able to map terms from the input domain onto a target ontology.

Based on the above conclusions, we can get an understanding of the role of domain adaptation for the task of mapping terms onto a target ontology from any domain.

7.0.0.1 Future Work

Further research is needed in the direction of the methods that we use for the task of domain adaptation of LLMs. For the MLM task, it can be tested if using an artificial 'SEP' token between the input pairs of sentences is beneficial for domain adaptation. We can also look into what would be the effect of using an artificial token instead of the 'SEP' token between the input sentences for fine-tuning via the MLM method. Furthermore, it can be tested if extending the MLM method to different aspects of the ontology, like using Masked Entitiy Modelling or Masked Relation Modelling, which would work on entities and relationships of the ontology, respectively, be beneficial methods for domain adaptation. For the classification task, it has to be tested if using different terms from the ontology, when paired and given an artificial label, acts as a better driving factor to improve domain adaptation via this task. Lastly, since for the MNR task, the hard negative sampling method yields better results instead of randomly sampled negatives, we can further investigate different methods for creating hard negative pairs to improve performance on the task of classifying terms from the input domain onto the target ontology. Also, we could see that the classification task did not perform well for our experiment setup. Further investigation can be carried out about this task to analyze why this task did not

[1] This average percentage point increase is computed over the map@5 soft threshold evaluation of both the English and the German language models.

work for our experiment setup, and it could be checked if changing the default hyperparameters can yield better performance via the classification method on the end task of mapping terms.

Glossary

Accuracy Accuracy is a metric that is used to assess how well a model correctly predicts the outcomes of a particular task against all the predictions made by the model.

Classification In NLP, classification tasks are formulated as supervised learning tasks6, wherein the task involves assigning predefined labels to a set of input sentences. Usually, classification tasks are used for solving problems like sentiment detection, language detection, etc. wherein we can have a set of pre-defined labels to input sentences and train a model, later asking it to classify sentences into one of the pre-defined categories.

ESCO Ontology ESCO (European Skills, Competences, Qualifications, and Occupations) is the European multilingual classification of Skills, Competences, and Occupations. ESCO works as a dictionary, describing, identifying, and classifying professional occupations and skills relevant to the EU labor market and education and training.

Gold Standard Dataset These are datasets that are used as a benchmark to perform the final evaluation on the end task.

Job ads Data Job ads data comprises terms in a job application that candidates use to apply for a job in a particular field.

Knowledge Graph In knowledge representation and reasoning, a knowledge graph is a knowledge base that uses a graph-structured data model or topology to integrate data.

LLMs Large Language models are trained on large datasets of text, such as articles, and websites, and use this data to learn patterns and relationships in language. These models are central to understanding what language is and how it is generated and can be adapted to different tasks by fine-tuning.

Mean Average Precision (mAP) This metric represents the number of relevant documents that are present in the top relevant results given by a particular model.

MLM Masked language modeling (MLM) is a semi-supervised language modeling task used for learning text representations for a particular domain. Training via MLM is done by masking words with a special token and making the model predict the masked words back.

MNR Multiple Negative Ranking (MNR) is a technique used for training models such that they get better at handling imbalanced class distributions.

MNR Loss The Multiple Negative Ranking Loss is a loss function that employs the MNR technique. It is an unsupervised learning method3, that takes pairs of positive and negatively related sentences to an anchor and optimizes their distances.

RDF Resource Description Framework is a standard for representing data on the web. It is a framework for expressing relationships between resources in a way that is machine-readable and can be easily processed by software applications.

SPARQL SPARQL is a query language used to query data in the RDF framework. It is similar in structure to the Structured Query Language (SQL), and the data is extracted in triplets.

TSV Tab-separated values is a format of file wherein the data in every row is stored with a separation of a tab (4 spaces) in between each entry.

URI Uniform Resource Identifier is a unique identifier for a particular resource on the web, used to identify a resource.

Validation Dataset This is a dataset that is used to check the performance of a particular model while it is training.

Validation Loss This is a loss that is computed with the validation dataset to assess the performance of a model.

XLM-RoBERTa This is a multi-lingual language model pre-trained on 2.5TB of filtered CommonCrawl data containing 100 languages.

References

A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL `https://aclanthology.org/2020.acl-main.747`.

J. Devlin, M.-W. Chang, K. Lee, and 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)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423`.

A.-s. Gnehm, E. Bühlmann, H. Buchs, and S. Clematide. 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, Nov. 2022a. Association for Computational Linguistics. URL `https://aclanthology.org/2022.nlpcss-1.2`.

A.-S. Gnehm, E. Bühlmann, and S. Clematide. Evaluation of transfer learning and domain adaptation for analyzing German-speaking job advertisements. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3892–3901, Marseille, France, June 2022b. European Language Resources Association. URL `https://aclanthology.org/2022.lrec-1.414`.

S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, July 2020. Association for Computational

Linguistics. doi: 10.18653/v1/2020.acl-main.740. URL `https://aclanthology.org/2020.acl-main.740`.

M. Henderson, R. Al-Rfou, B. Strope, Y.-h. Sung, L. Lukacs, R. Guo, S. Kumar, B. Miklos, and R. Kurzweil. Efficient natural language response suggestion for smart reply, 2017. URL `https://arxiv.org/abs/1705.00652`.

B. Kim, T. Hong, Y. Ko, and J. Seo. Multi-task learning for knowledge graph completion with pre-trained language models. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1737–1743, Barcelona, Spain (Online), Dec. 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.153. URL `https://aclanthology.org/2020.coling-main.153`.

D. Li, S. Yang, K. Xu, M. Yi, Y. He, and H. Wang. Multi-task pre-training language model for semantic network completion, 2022. URL `https://arxiv.org/abs/2201.04843`.

J. Li, C. Tao, W. Wu, Y. Feng, D. Zhao, and R. Yan. Sampling matters! an empirical study of negative sampling strategies for learning of matching models in retrieval-based dialogue systems. 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 1291–1296, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1128. URL `https://aclanthology.org/D19-1128`.

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 1 edition, 2008.

S. Mirhosseini, G. Zuccon, B. Koopman, A. Nguyen, and M. Lawley. Medical free-text to concept mapping as an information retrieval problem. In *Proceedings of the 2014 Australasian Document Computing Symposium*, ADCS '14, page 93–96, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450330008. doi: 10.1145/2682862.2682880. URL `https://dl.acm.org/doi/10.1145/2682862.2682880`.

L. Tunstall, N. Reimers, U. E. S. Jo, L. Bates, D. Korat, M. Wasserblat, and O. Pereg. Efficient few-shot learning without prompts. *arXiv preprint arXiv:2209.11055*, 2022. URL `https://doi.org/10.48550/arXiv.2209.11055`.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the*

31st International Conference on Neural Information Processing Systems, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

M. Zhang, K. N. Jensen, and B. Plank. 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, June 2022. European Language Resources Association. URL `https://aclanthology.org/2022.lrec-1.46`.