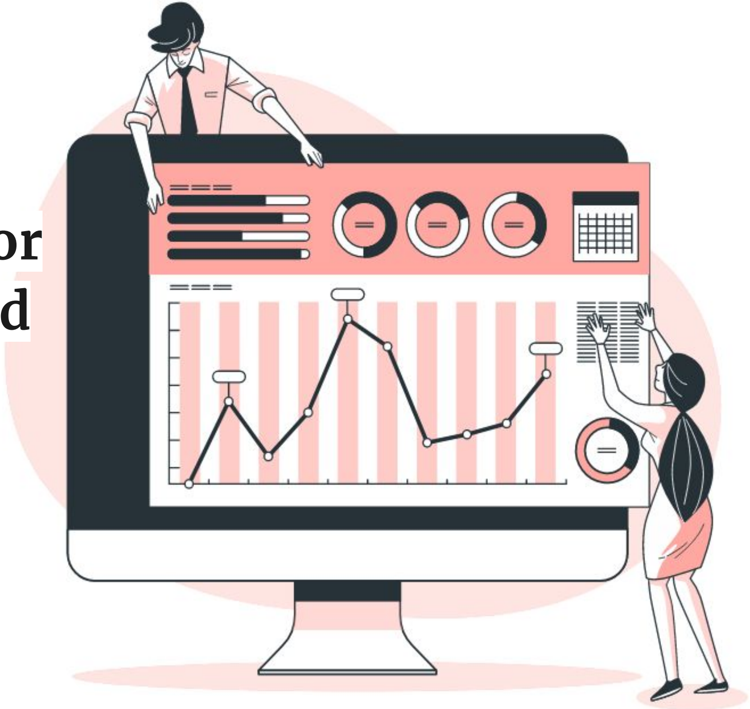
universidade de aveiro deti departamento de eletrónica, telecomunicações e informática

Development of a computational system for determining ESCO competences associated with training offers

Dissertation / Internship Preparation

Teacher - Augusto Silva
Supervisor - António Neves
Co-Supervisor - Fabianne Ribeiro
2023/2024

Artur Romão, 98470



Contents

01 Introduction

02 Taxonomies for Classification of Occupations and Skills

03 Large Language Models

04 Applying LLMs and NLP to Skill Taxonomies

05 Methodology

01

Introduction

Introduction

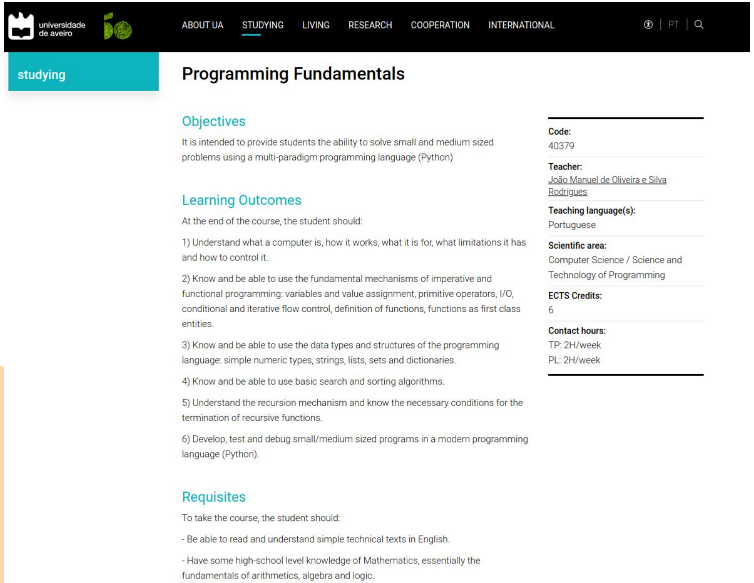
Continuous evolution of knowledge raised the importance of lifelong learning for up-skill and re-skill (ex. micro-credentials)

Challenges of Business Sector in finding training offers to fill the gaps in the workforce skills and in linking them to the effective acquisition of knowledge

To address these problems, the European Commission created the **European Taxonomy for Skills, Competences, Qualifications and Occupations (ESCO)**

ESCO provides an **API** however, it is not capable of accurately mapping training offers to taxonomy skills

UA’s Educational Offer Organization



Programming Fundamentals

Objectives

It is intended to provide students the ability to solve small and medium sized problems using a multi-paradigm programming language (Python)

Learning Outcomes

At the end of the course, the student should:

1) Understand what a computer is, how it works, what it is for, what limitations it has and how to control it.

2) Know and be able to use the fundamental mechanisms of imperative and functional programming: variables and value assignment, primitive operators, I/O, conditional and iterative flow control, definition of functions, functions as first class entities.

3) Know and be able to use the data types and structures of the programming language: simple numeric types, strings, lists, sets and dictionaries.

4) Know and be able to use basic search and sorting algorithms.

5) Understand the recursion mechanism and know the necessary conditions for the termination of recursive functions.

6) Develop, test and debug small/medium sized programs in a modern programming language (Python).

Requisites

To take the course, the student should:

- Be able to read and understand simple technical texts in English.

- Have some high-school level knowledge of Mathematics, essentially the fundamentals of arithmetics, algebra and logic.

Code:
40379

Teacher:
João Manuel de Oliveira e Silva Rodrigues

Teaching language(s):
Portuguese

Scientific area:
Computer Science / Science and Technology of Programming

ECTS Credits:
6

Contact hours:
TP: 2H/week
PL: 2H/week

Example of a DPUC for the course of Programming Fundamentals
Available in: https://www.ua.pt/en/uc/12286

- University of Aveiro’s (UA) educational offer is available as Pedagogical Dossiers (DPUCs) which work as ID cards of UA’s courses and micro-credentials
- DPUCs contain many fields related to the course, such as: **name**, **contents**, **learning outcomes**, **requirements**, **assessment**, …
- However, it is not worth to use all of these fields in order to obtain a list of ESCO skills

Objectives

- Implementation and testing of a system to manage the skills of UA's educational offer and to match them to ESCO skills

- Development of a pipeline that connects the ESCO framework (API), UA's DPUCs and a Large Language Model (LLM) framework

- Testing LLM frameworks and Natural Language Processing (NLP) libraries to integrate into the system and to help mapping the training offer to ESCO skills

- Deploy the system to automate the process of obtaining ESCO skills

- Evaluate the system's performance using manual verification of skills by course directors

Main goal - Provide UA's academic community with a platform that helps:

- Current and future students to understand better the educational offer
- Human Resources representers to recognize former students' skills upon hiring them

02

Taxonomies for Classification of Occupations and Skills

Taxonomies for Classification of Occupations and Skills

Taxonomies are structured databases that work as **dictionaries** to categorize and describe occupations and skills in the labour market.

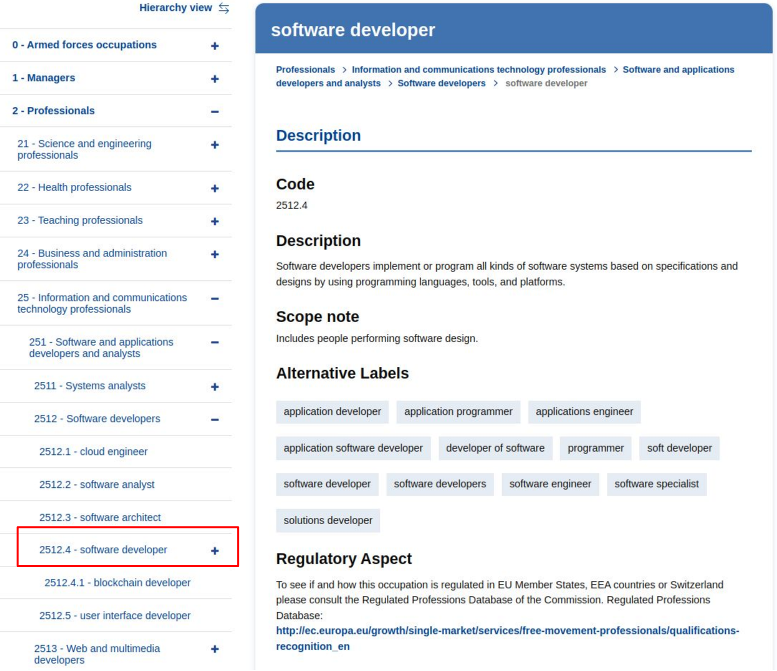
From their multiple advantages, the following must be highlighted:

Standardization and clarity

Professional development and guidance

Recruitment and human resources management

ESCO



Hierarchy view ⇆

0 - Armed forces occupations +

1 - Managers +

2 - Professionals −

21 - Science and engineering professionals +

22 - Health professionals +

23 - Teaching professionals +

24 - Business and administration professionals +

25 - Information and communications technology professionals −

251 - Software and applications developers and analysts −

2511 - Systems analysts +

2512 - Software developers −

2512.1 - cloud engineer

2512.2 - software analyst

2512.3 - software architect

2512.4 - software developer +

2512.4.1 - blockchain developer

2512.5 - user interface developer

2513 - Web and multimedia developers +

software developer

Professionals > Information and communications technology professionals > Software and applications developers and analysts > Software developers > software developer

Description

Code

2512.4

Description

Software developers implement or program all kinds of software systems based on specifications and designs by using programming languages, tools, and platforms.

Scope note

Includes people performing software design.

Alternative Labels

application developer | application programmer | applications engineer | application software developer | developer of software | programmer | soft developer | software developer | software developers | software engineer | software specialist | solutions developer

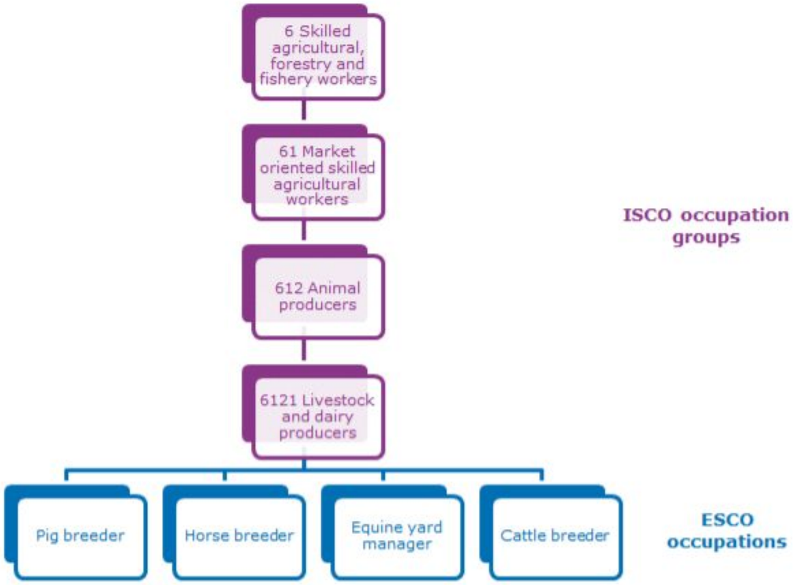
Regulatory Aspect

To see if and how this occupation is regulated in EU Member States, EEA countries or Switzerland please consult the Regulated Professions Database of the Commission. Regulated Professions Database: **http://ec.europa.eu/growth/single-market/services/free-movement-professionals/qualifications-recognition_en**

Example of an ESCO's Web Service page
Source: ESCO - Software Developer

- Taxonomy to describe, identify and classify skills, qualifications and occupations that are relevant to the European Union (EU) labour market
- ESCO is currently available in 28 languages
- Describes 3,008 occupations and 13,890 skills
- Follows a hierarchical approach according to ISCO-08
- Provides 2 APIs - a user-friendly **Web Service** and a **Local Version** that can be downloaded and integrated in information systems

ISCO

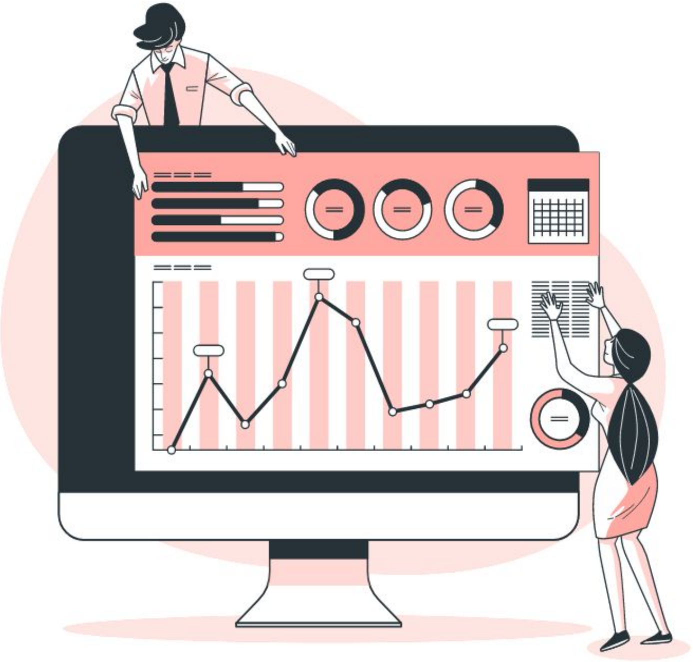

6 Skilled agricultural, forestry and fishery workers
61 Market oriented skilled agricultural workers
612 Animal producers
6121 Livestock and dairy producers
Pig breeder
Horse breeder
Equine yard manager
Cattle breeder
ISCO occupation groups
ESCO occupations

Role of ISCO-08 in the hierarchical structure of the ESCO occupations pillar
Source: ESCO - International Standard Classification of Occupations (ISCO)

- International taxonomy to classify occupations
- Developed by the International Labour Office (ILO)
- Two most recent versions are ISCO-88 (1988) and ISCO-08 (2008)
- Inspired ESCO because of its hierarchical structure
- ISCO's incorporation in ESCO allows compatibility and interoperability between both systems

03

Large Language Models (LLMs)



Large Language Models (LLMs)

LLMs are:

- Neural Networks with billions of parameters
- Trained on vast quantities of unlabelled data (to understand language patterns, grammar and semantics) and labelled data (to guide the model towards more specific tasks)
- Dependent on Natural Language Processing (NLP), since it handles tasks such as tokenization, part-of-speech tagging and entity recognition, essential for training LLMs effectively
- Powerful frameworks for text processing with role-playing ability and human language comprehension
- An integral part of a wide range of applications (ex. *chatbots*, translators, content recommendation systems, …)

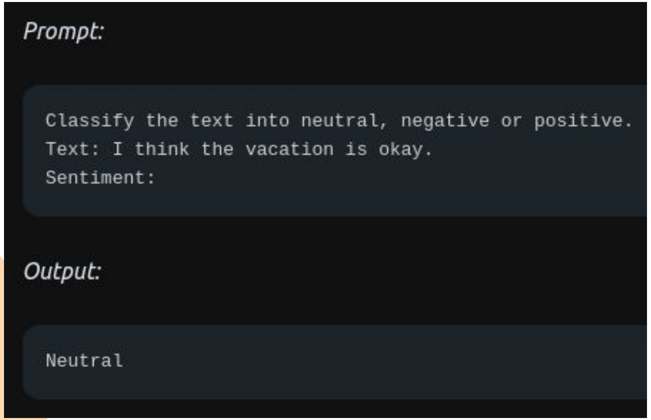
04

Applying LLMs and NLP to Skill Taxonomies

Prompt Engineering

The concern in developing and optimizing prompts to achieve the best possible answers gave rise to a new discipline, **Prompt Engineering**.

It gathers the set of techniques to enhance the interactions with LLMs, allowing for their augmentation with domain knowledge and integration with external tools.



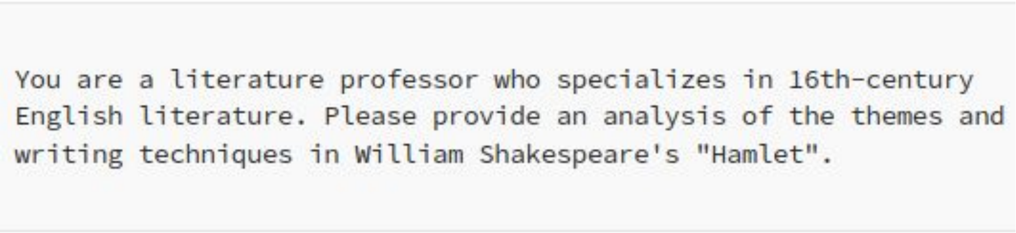
Prompt:

```
Classify the text into neutral, negative or positive.
Text: I think the vacation is okay.
Sentiment:
```

Output:

```
Neutral
```

Example of Zero-Shot Prompting
Extracted from: Zero-Shot Prompting

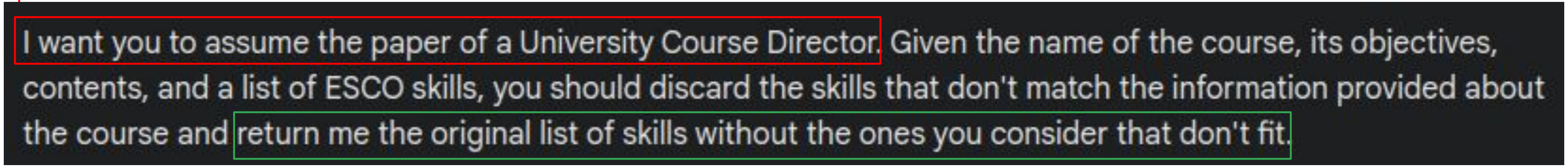


```
You are a literature professor who specializes in 16th-century
English literature. Please provide an analysis of the themes and
writing techniques in William Shakespeare's "Hamlet".
```

Example of Role-Play Prompting
Adapted from: Prompt Engineering — Role Prompting

Combining Zero-Shot and Role-Playing

Role-Play Prompting - The LLM is asked to take on the role of a Course Director



I want you to assume the paper of a University Course Director. Given the name of the course, its objectives, contents, and a list of ESCO skills, you should discard the skills that don't match the information provided about the course and return me the original list of skills without the ones you consider that don't fit.

Zero-Shot Prompting - The LLM is asked to perform a task on data it wasn’t trained on

05

Methodology

Methodology

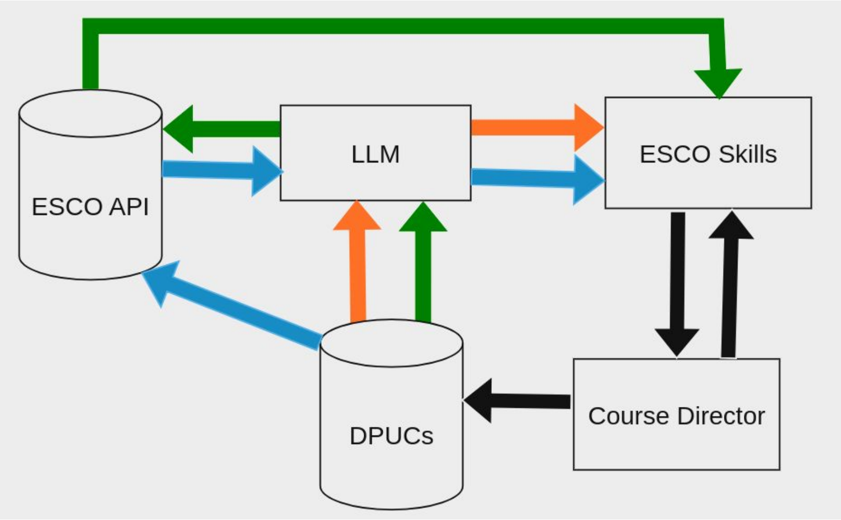


Diagram representing the system's pipeline

- 5 "entities" - a document containing all DPUCs and micro-credentials, ESCO API, an LLM framework, the course directors and the ESCO Skills, which are the final product

- Three different approaches illustrated: **orange**, **blue** and **green**

Regarding implementation:

- Extracting the desired fields from the DPUCs document (course **name**, **contents** and **learning outcomes**)

- Programmatic **integration of an LLM** (ex. *Google Bard* through web cookies)

- **ESCO's API** queries are done through HTTP requests

- Final **assessment** with Course Directors

Gantt Diagram

Task	Months (2023/2024)									
	Sept	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun
Write documentation	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧
Identify research question	🟦									
Research into ESCO and related skills taxonomies	🟨	🟨								
Install ESCO API and read its documentation		🟫								
State of the Art and Literature Review		🟪	🟪							
Search for NLP and LLM frameworks to integrate programmatically into the system		🟩	🟩							
Conduct tests and experiments with LLMs using DPUCs			🟪	🟪	🟪	🟪				
Development of each component of the system pipeline					🟩	🟩				
Integrate all the components to automate the system flow							🟦			
Deploy the system pipeline							🟥	🟥		
Test and assessment with course directors									🟩	