MICRO IDEA

MICRO-credentials
Identifying,
DEveloping, testing and
Assessing innovative approaches

Project number: 101132889 - MICROIDEA - ERASMUS-EDU-2023-PI-FORWARD

One skills in demand analysis tool/Platform

WP 1 | Activity 1.4

Developed by the University of Peloponnese | July, 2025

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Table of Contents

A. Executive Summary

The D1.4 deliverable for Work Package 1 (WP1) of the MICROIDEA project, 'Development of an Online Tool for Occupation and Skills Extraction,' enhances our understanding of labor market dynamics through technological advances. This report summarizes our research and development efforts to create an advanced tool using machine learning (ML) and natural language processing (NLP). The primary goal of this tool is to extract, analyze, and present occupation and skills data from online job postings, providing detailed insights into labor market trends and skill demands.

Developed by the Data & Media Laboratory (DM Lab) within the Department of Electrical and Computer Engineering at the University of Peloponnese, this tool marks a significant advancement over its offline predecessor. Initially, it aggregated and processed job postings from various online job vacancy (OJV) portals, performing tasks such as data cleansing and deduplication to produce a coherent dataset. The offline version laid a strong foundation by demonstrating how automated techniques can identify critical employment trends from diverse data sources.

Moving to an online platform significantly enhances accessibility and functionality. By integrating state-of-the-art LLMs, the tool has expanded its analytical capabilities, particularly in text understanding and generation, essential for interpreting complex job data. The use of LLMs enables the tool to manage a wide range of text-based information, facilitating the extraction of industry-specific details according to the NACE 2.0 classification.

Furthermore, the tool employs advanced text embeddings, a key feature of modern NLP, to effectively implement similarity matching functions. These embeddings allow the system to accurately match job descriptions with relevant categories within the ISCO-08 and ESCO skills taxonomies. This capability is crucial in parsing and categorizing occupations and skills from extensive datasets of unstructured text found in online job postings. By converting text data into a form that can be geometrically analyzed, text embeddings enable the tool to efficiently detect similarities and differences in job-related content.

These enhancements allow for dynamic, real-time analysis of labor market data. Stakeholders now have access to actionable insights with unprecedented speed, facilitating timely and informed decision-making. The application of LLMs and text embeddings not only enriches the tool's functionality but also enhances its precision in dissecting and interpreting the labor market landscape. This evolution represents a significant step forward in the tool's ability to support the strategic needs of policymakers, educators, and industry leaders in adapting to rapidly changing skill demands.

An essential aspect of this project within the MICROIDEA framework is its focus on scalability and adaptability. The tool is designed to meet the immediate needs of the Greek, Cypriot, and Spanish labor markets and envisioned as a scalable model that can be adapted for broader European and global contexts. This adaptability is crucial as it allows for the extension of the tool's capabilities to other regions and sectors, potentially benefiting a wider audience and addressing the universal need for precise labor market analytics.

Furthermore, this project supports the MICROIDEA initiative's goal to foster innovation in educational and occupational practices. By providing a detailed analysis of current and emerging skill requirements, the tool aids educational institutions and policymakers in developing targeted training programs that are in sync with market needs. This alignment is crucial for addressing skill gaps and enhancing workforce employability, thus contributing to economic growth and competitiveness.

B. Introduction

Work Package 1 (WP1) of the MICROIDEA project establishes the foundational infrastructure for a robust system aimed at analyzing labor market trends and guiding the development of micro-credentials. Central to WP1 is the creation of an AI-powered online tool that extracts and structures data from job postings, focusing on occupations and skills relevant to dynamic labor market demands.

The **Online Skills-in-Demand Analysis Tool** is available through the main MICROIDEA portal at https://portal.micro-idea.eu/. To access it, navigate to the **"Navigate"** menu and select **"Dashboards".**

Deliverable D1.4 operationalizes the goals of WP1 by delivering a fully integrated, scalable online system for occupation and skills extraction. Building on the theoretical and technical groundwork set by D1.1 through D1.3, this deliverable consolidates prior insights into a practical, web-based application. It leverages advanced AI and NLP techniques—such as automated data crawling, multilingual translation, normalization, and semantic metadata structuring—aligned with established international classifications like ESCO, ISCO-08, and NACE Rev. 2.

While earlier efforts focused on pipeline design and validation, D1.4 represents a shift toward deployment and real-time capability. The tool targets job postings from Greek, Cypriot, and Spanish portals, providing a foundation for region-specific analytics while being architected for future extensibility. It plays a critical role in enabling downstream activities in MICROIDEA, particularly in WP2 and WP4, where labor insights inform training design, micro-credentialing strategies, and policy recommendations—especially in sectors like tourism, where skill needs evolve rapidly.

By aligning technological development with strategic policy and educational goals, D1.4 advances MICROIDEA's mission to bridge the gap between labor market intelligence and responsive learning pathways.

C. Glossary of Terms

Term	Definition
AI (Artificial Intelligence)	A broad field of computer science focused on building systems capable of performing tasks that typically require human intelligence, such as language understanding, pattern recognition, and decision-making.
API (Application Programming Interface)	A set of protocols and tools that allows different software systems to communicate and exchange data.
ChromaDB	A lightweight vector database used to store and query high-dimensional embeddings
Embedding (Text Embedding)	A technique that converts text into numerical vectors in a high-dimensional space, enabling semantic comparison of phrases or documents.
ESCO (European Skills, Competences, Qualifications and Occupations)	European multilingual classification of Skills, Competences and Occupations
Ground Truth Dataset	A manually verified dataset used as a benchmark for evaluating the accuracy and reliability of AI outputs.
HTML (HyperText Markup Language)	The standard language for formatting content on the web.
ISCO-08 (International Standard Classification of Occupations 2008)	A classification system maintained by the International Labour Organization (ILO) that categorizes occupations into a four-level hierarchical structure
ISCED (International Standard Classification of Education)	A framework developed by UNESCO to classify educational programs and qualifications across different countries into standardized levels.
JSON (JavaScript Object Notation)	A lightweight data format used for representing structured information.
LLM (Large Language Model)	A type of deep learning model trained on vast text corpora to understand, generate, and analyze natural language.

LLM-as-a-Judge	An evaluation framework in which a language model is used to assess the quality of AI-generated outputs, simulating human judgment.
MariaDB	An open-source relational database system.
NACE Rev. 2 (Statistical Classification of Economic Activities)	The European Union's standard for classifying economic activities, used to categorize job postings by industry sector.
NER (Named Entity Recognition)	A subtask of information extraction that identifies proper names or phrases (entities) in text.
Ollama	A tool used to run large language models locally on private infrastructure, allowing for secure and fast inference without external API calls
Power BI	A Microsoft data visualization tool used in MicroIDEA to create interactive dashboards
Prompt Engineering	The design of specific instructions or templates (prompts) to guide the behavior and output of LLMs
Pydantic	A Python data validation library used to define the structure and types of metadata extracted from job postings
RAG (Retrieval-Augmented Generation)	A hybrid AI architecture that retrieves relevant documents from a database and uses them to guide a language model's response, improving contextual accuracy.
Temperature	parameter that controls the randomness and creativity of the model's output
Transformer (Architecture)	A deep learning architecture based on self-attention mechanisms, which underpins the operation of LLMs.
Vector Space / Vector Representation	A mathematical space in which text inputs are represented as vectors to facilitate similarity comparison, clustering, or classification.

D. Technological and Research Landscape

The recent breakthroughs in Artificial Intelligence (AI) and Natural Language Processing (NLP) have profoundly reshaped how unstructured text data is processed, analyzed, and utilized. Central to this transformation is the emergence of LLMs such as BERT, GPT-3, and GPT-4, all of which are built upon the Transformer architecture. These models represent a new generation of AI systems capable of understanding and generating human-like text with remarkable accuracy. Their success lies in their capacity to model linguistic patterns, semantic relationships, and contextual dependencies using self-attention mechanisms that evaluate the significance of each word in relation to the entire input sequence [1].

LLMs are trained on massive multilingual and multi-domain datasets, equipping them with a deep understanding of language use across sectors, registers, and cultures. Unlike earlier rule-based or statistical NLP approaches, Transformer-based models can infer meaning beyond surface-level word co-occurrence, capturing subtle nuances and varied formulations of the same concept [2]. This is particularly valuable in labor market contexts, where job postings and CVs are often written informally, vary in format and terminology, and differ widely by country, industry, or employer.

In this evolving landscape, LLMs offer exceptional capabilities for structuring job-related data at scale. When applied to job advertisements, these models can extract detailed metadata—such as job titles, required and optional skills, qualifications, employment type, experience levels, and even implicit details like soft skills or tone—by interpreting the content holistically. This extraction process is not based on rigid patterns but rather on contextual understanding, making it possible to deal with heterogeneous or noisy inputs from multiple sources and languages [3][4]. The extracted information is then normalized against international taxonomies such as ISCO-08[1] for occupations, ESCO[2] for skills, NACE Rev. 2[3] for industry classification and ISCED[4] level for the classification of education level, ensuring consistency and comparability across the dataset.

Beyond the task of entity recognition and classification, LLMs play a critical role in similarity analysis. By embedding text into high-dimensional vector spaces, using models such as Sentence-BERT or text embedding engines, the system can determine the semantic proximity between a job description and predefined taxonomy concepts [5][6]. This allows for precise alignment even when terminology differs—such as recognizing that “customer service associate” and “retail front-desk agent” refer to the same occupational category. This capability is particularly important for tools like the one developed in MICROIDEA, which must handle multilingual, domain-specific, and unstructured data across national contexts.

Furthermore, the use of LLMs is transforming how job recommendation systems are designed and delivered. Traditional matching mechanisms rely heavily on keyword matching, Boolean filters, or predefined categories, often leading to suboptimal user experiences. In contrast, Transformer-based systems construct rich semantic profiles of both candidates and job postings, taking into account historical behavior, stated preferences, career trajectories, and inferred aspirations [7]. By learning from user interaction patterns—such as job views, saved searches, or application history—these models can offer highly personalized and adaptive recommendations. Such systems are capable of suggesting opportunities that align not only with a user’s formal qualifications but also with their potential for growth and career development.

Another important dimension introduced by LLMs is the ability to generate human-like text in real time. In recruitment and educational platforms, this enables more engaging and informative

[1] https://www.ilo.org/publications/international-standard-classification-occupations-2008-isco-08-structure

[2] https://esco.ec.europa.eu/en

[3] https://ec.europa.eu/eurostat/web/nace

[4] https://ec.europa.eu/eurostat/statistics-explained/index.php?title=International_Standard_Classification_of_Education_(ISCED)

interactions. Personalized job descriptions can be automatically adapted based on a user's profile. Feedback on applications or skill gaps can be generated with empathetic and actionable tone [8]. Career guidance and upskilling suggestions can be communicated in natural language, rather than static checklists. This capacity to produce tailored, context-aware, and linguistically fluent outputs enhances user experience and trust, contributing to greater platform engagement and retention.

These advances are not purely technological—they are strategic. Within the MICROIDEA framework, the deployment of Transformer-based LLMs underpins the transition from static, manually curated labor market data to dynamic, scalable, and multilingual analytics. The online tool for occupation and skills extraction leverages these models to process large volumes of job postings from Greece, Cyprus, and Spain, offering stakeholders real-time insights into evolving workforce needs. By bridging the gap between raw data and actionable intelligence, the system supports the design of targeted micro-credentials, informs national and regional policy, and enables more inclusive and responsive education and employment ecosystems.

Ultimately, this technological foundation ensures that the MICROIDEA project not only meets current analytical needs but is also adaptable to future labor market shifts. As job content continues to evolve and new skill demands emerge, LLM-powered systems provide the flexibility, accuracy, and scalability required to support evidence-based decisions across the labor market, education, and vocational training domains.

E. Methodology – System Components and Analytical Workflow

This section presents the methodological framework that underpins the transformation of unstructured job postings into structured, taxonomy-aligned labor market intelligence. The MicroIDEA platform employs a modular, AI-driven pipeline that integrates advanced Natural Language Processing techniques with robust data engineering workflows. Central to this approach is the use of Transformer-based LLMs, which enable the extraction and semantic interpretation of job-related metadata from diverse, multilingual sources. By leveraging the contextual reasoning capabilities of them, the system handles inconsistencies in language, structure, and terminology across job postings. These outputs are then normalized against international classification standards (ISCO, ESCO, NACE, ISCED), ensuring cross-country comparability and supporting scalable analytics for labor market monitoring, skills intelligence, and policy design.

This methodology is structured around five core components (Figure 1)**: (1) Data Acquisition and Metadata Identification, (2) Preprocessing and Semantic Cleaning, (3) LLM-based Entity Extraction, (4) Semantic Normalization using RAG and LLM Reranking**, and **(5) Evaluation and Iterative Optimization**. Each component builds upon the previous stage, ensuring both data integrity and analytical utility across the pipeline.

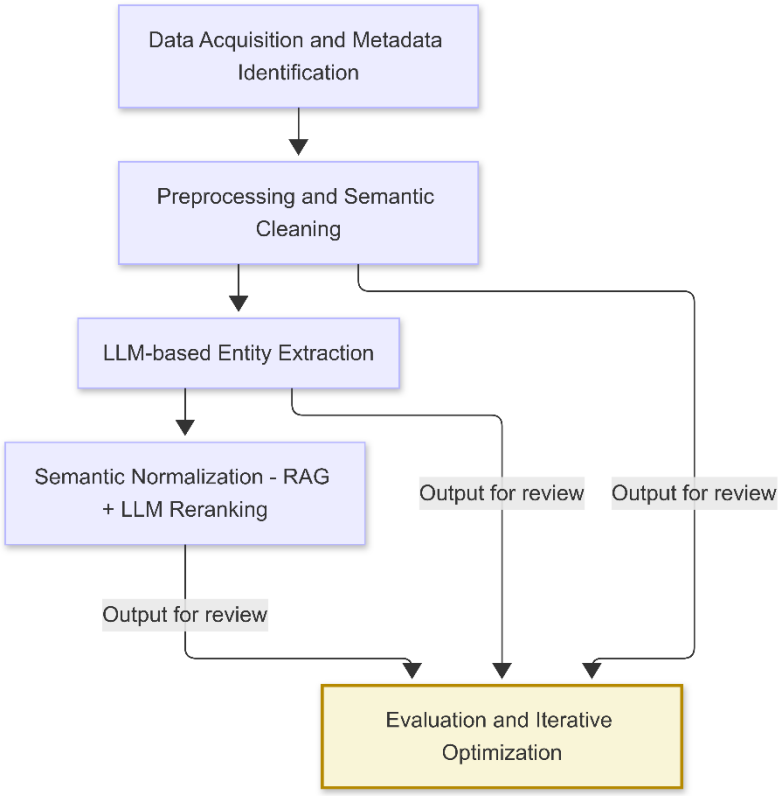

Figure 1: High-level pipeline of the MicroIDEA system

E.1 Data Acquisition and Metadata Field Identification

The process begins with the acquisition of job postings from national and regional job portals, such as those of Greece, Cyprus, and Spain. A custom web crawler operates on a scheduled basis to fetch postings directly from HTML pages or JSON-based APIs, capturing full job content as well as any structured metadata exposed by the portals.

Each record is ingested along with its provenance metadata: source URL, timestamp, job portal identifier, and country tag. This enables accurate tracking, versioning, and cross-country comparison of job-related data. A set of core metadata fields is identified during this stage—such

as job title, job description, location, company name, and date of publication—forming the minimal schema required for downstream enrichment and entity extraction. This minimal schema is stored in a relational database, serving as the backbone of the pipeline.

E.2 Preprocessing: Cleaning, Translation, and Semantic Deduplication

Once the job postings are collected from the target portals, they enter a critical preprocessing phase aimed at transforming raw, heterogeneous text into a coherent and semantically aligned format suitable for AI-based extraction. This phase involves several tightly connected operations—text cleaning, translation, and deduplication—each of which plays a vital role in ensuring data quality, consistency, and readiness for downstream processing.

The first task is text cleaning. Job advertisements retrieved from HTML-based sources often contain a range of non-informative content such as markup tags, styling characters, encoding errors, and other forms of visual or structural noise. These elements are systematically removed through automated scripts, resulting in clean, plain-text content that preserves the semantic core of the original advertisement. This cleaned version is stored alongside the raw version in the database to allow for later verification, fallback strategies, or reprocessing if needed.

Since MicroIDEA operates across countries with different official languages—Greek, Spanish, and English—translation into a shared language is essential for consistent processing. Their understanding of nuanced job-related language in Greek or Spanish is often weak. English is used as the intermediate language, while it aligns with the training data of most LLMs, improving the quality of downstream extraction [9]. Non-English job titles and descriptions are translated using neural machine translation, either via APIs or local LLMs, depending on privacy and latency requirements. Both original and translated versions are stored to support transparency and bilingual auditing (e.g., `job_description_original` and `job_description_en`).

A critical step in preprocessing component is semantic deduplication, addressing the widespread issue of job postings being reposted or cross-listed with slight variations [10]. The presence of duplicates has a significant impact on data integrity and labor market analysis. These duplicates introduce biases into the data analysis, resulting in misleading conclusions about employment trends and the demand for specific skills [11].

To prevent distortion in statistical analysis, MicroIDEA uses a two-stage approach. First, vector embeddings are generated for key fields—such as location, title, company, and description—to identify semantically similar entries. Then, candidate pairs are assessed by a large language model, which determines whether they refer to the same job and evaluates their degree of difference. The system uses metadata fields such as `is_duplicate` and `duplicate_group_id` to store results, keeping one main version and linking to duplicates for tracking and long-term analysis.

Together, these preprocessing steps form the bridge between raw web-scraped data and high-quality, semantically aligned content. By cleaning, translating, and deduplicating the postings in a structured and traceable manner, the MicroIDEA pipeline ensures that downstream AI models operate on consistent, language-neutral, and duplication-free input—laying a robust foundation for meaningful metadata extraction and labor market intelligence.

E.3 LLM-Based Structured Metadata Extraction

Metadata extraction from job postings is a complex task due to the unstructured, inconsistent, and domain-specific language commonly used by recruiters. Accurately identifying key entities—such as required skills, qualifications, and job titles—is essential for bridging the gap between labor supply and demand, enabling effective job matching, skill gap analysis, and career guidance [12].

This component forms the semantic backbone of the MicroIDEA pipeline, responsible for transforming preprocessed job postings into structured, taxonomy-aligned metadata. Leveraging the capabilities of Transformer-based LLMs, it enables accurate and context-aware extraction from diverse, multilingual, and often informal job texts.

Guided by the predefined JPInfo schema (Figure 2), the component prompts LLMs to extract a rich set of fields—such as `job title`, `hiring_company`, `location`, `required skills`, `qualifications`, and `benefits`. Unlike traditional NER methods, the LLM-based approach can infer implicit information and disambiguate semantically similar elements, such as distinguishing a degree from a soft skill or a job function from a formal title [13].

The component is designed to be robust across domains and formats, with prompt adjustments enabling enhanced extraction for sector-specific roles (e.g., technical, healthcare, tourism). Outputs are returned in JSON and automatically validated for structural and semantic integrity before proceeding to the normalization stage [14].

```
class JPInfo(BaseModel):
advertising_company: str = Field(..., description="The name of the company or
agency responsible for managing the recruitment process. This may be different from
the seeking company, particularly if an external recruitment agency is used.")
hiring_company: str = Field(..., description="The name of the company that is
actively seeking to fill the position. This is the organization where the successful
candidate will be employed.")
location_city_or_area: str = Field(..., description="Specific city or area of the
job location")
job_title: str = Field(..., description="The official title of the position as
described in the job posting. This should clearly represent the specific role and
level within the organization, including any relevant specializations or focus
areas. Examples include 'Senior Data Scientist,' 'Marketing Manager,' or 'Junior
Software Developer specializing in Frontend Development.'")
occupation: str = Field(..., description="The broader category or profession to
which the job belongs. This should generalize the role, focusing on the type of work
performed rather than specific titles. For example, for 'Senior Machine Learning
Engineer,' the occupation could be 'Software Engineer' or 'Machine Learning
Specialist.' If the occupation cannot be determined, default to using the job title
as a fallback option.")
job_season: str = Field(..., description="If applicable, the time period or season
for the job.")
```

Figure 2: Predefined JPInfo Schema

By converting unstructured text into standardized, machine-interpretable records, this component enables scalable analytics, cross-country comparisons, and the development of intelligent labor market tools across the MicroIDEA ecosystem.

E.4 Semantic Normalization via RAG and LLM Reranking

Once structured metadata has been extracted from each job posting, the next crucial step in the MicroIDEA pipeline is semantic normalization. This phase ensures that the diverse, free-text content obtained during extraction is mapped to standardized concepts, enabling reliable aggregation, cross-country comparisons, and integration with European labor market classification taxonomies such as ESCO (European Skills, Competences, Qualifications and Occupations), ISCO-08 (International Standard Classification of Occupations), NACE Rev. 2 (economic activities), and ISCED (education levels).

Job postings typically exhibit significant linguistic variability—even when referring to the same underlying concepts. For example, the occupation “sales associate” might also appear as “retail advisor” or “store clerk,” depending on the employer or country. Similarly, a skill like “teamwork” could be phrased as “ability to work collaboratively” or “cooperation with colleagues.” Without a normalization step, such variation would fragment the dataset and weaken any attempt to produce coherent insights or statistics.

To address this, MicroIDEA employs a hybrid approach that combines fast similarity-based retrieval with deep semantic reasoning. The process begins by embedding each free-text entity—such as a

job title, skill, or industry category—into a high-dimensional vector space using transformer-based text embedding models. This embedding represents the semantic meaning of the text and enables comparison with pre-embedded entries in curated lexicons derived from taxonomies. Using a vector database (ChromaDB), the system retrieves the top-k most similar standardized entries for each extracted term [15].

However, rather than relying solely on numerical similarity scores, the pipeline incorporates a second stage to enhance precision and interpretability. The list of candidate matches is passed to a local LLM, which acts as a semantic reranker. This model is prompted to evaluate the contextual meaning of the original term and determine which of the retrieved taxonomy entries most accurately represents it. The LLM may also explain why it selected a particular match over others, allowing for greater transparency and the possibility of auditing borderline cases. This two-step strategy ensures that subtle nuances are not overlooked—for example, that "Data Engineer" is matched to the correct occupational unit rather than "IT Technician," even if both have similar wording.

Once a final match is selected, the system records the standardized value in a dedicated normalized field—for example, `job_title_esco`, `skill_esco`, `nace_code`, or `isced_level`. Additional metadata may also be stored, including a confidence score, a classification of skill type (e.g., technical or transversal), and a flag indicating whether the term appears to be new or underrepresented in the taxonomy (e.g., `is_new_skill` = True). These additional annotations support further research, taxonomy extension, and dynamic updating of classification schemes.

Normalization is applied not only to isolated terms but also to more complex lists—such as arrays of required skills or qualifications. In such cases, the embedding and reranking processes are performed iteratively for each list item.

By grounding extracted entities in structured, interpretable vocabularies, this normalization step enables robust, high-quality analytics. It supports aggregation of postings by occupation, industry sector, location, skill group, and educational level, allowing MicroIDEA to deliver insights into trends such as rising demand for specific digital competencies, sectoral skill shortages, or geographic mismatches in workforce qualifications. In turn, these outputs inform micro-credential development, curriculum alignment, and targeted policy interventions at regional, national, and European levels.

Semantic normalization thus acts as a bridge between unstructured real-world job content and standardized analytical frameworks, ensuring that the MicroIDEA platform delivers consistent, comparable, and actionable labor market intelligence across linguistic, geographic, and sectoral boundaries.

E.5 Evaluation and Iterative Optimization

The evaluation component plays a pivotal role in validating, refining, and maintaining the quality of the MicroIDEA methodology. Given the pipeline's reliance on AI-driven processes—such as LLM-based extraction, semantic deduplication, and taxonomy normalization—rigorous, multi-level evaluation is essential to ensure accuracy, consistency, and adaptability across components. The foundation of this component lies in a set of carefully curated ground truth datasets. These include:

- **Manually annotated job postings** with normalized metadata fields to evaluate extraction accuracy.
- **Deduplication benchmarks**, featuring real-world cases of near-identical postings across platforms.
- **Translation test sets**, highlighting linguistically complex or domain-specific examples to assess robustness.

These resources serve as the basis for evaluating each core component using the *LLM-as-a-Judge* approach. This approach treats LLMs not only as generators but also as evaluators. Rather than a single metric, it's a flexible framework designed to approximate human judgment. This method is adaptable to each specific application, with its effectiveness depending on the evaluation prompt, the chosen model, and the complexity of the task. Research has shown that LLM-as-a-Judge often aligns well with human preferences [16].

Given multiple system-generated outputs—e.g., from alternative prompt designs, extraction models, or translation variants—an independent LLM is prompted to assess accuracy, completeness, and semantic fidelity. Evaluation instructions are guided by scoring rubrics tailored to each task (e.g., skill relevance, proper title mapping, contextual accuracy in normalization).

Once benchmarked against ground truth, the same LLM-as-a-Judge framework is used to test and compare various configurations of the pipeline. For instance, it enables the comparison of commercial vs. open-source LLMs (e.g., GPT-4 vs. LLaMA 3), different deduplication thresholds, or taxonomy mapping strategies. This supports data-informed decisions about trade-offs between performance, cost, latency, and privacy across deployment scenarios.

Beyond qualitative judgment, automated rule-based checks are also embedded to ensure schema conformity, data completeness, and type validation. Outputs that fail validation are flagged for fallback handling or manual review. All evaluation data—judgments, scores, and flagged issues—are logged in a dedicated database layer. This audit trail supports traceability, quality assurance, and continuous improvement. It also enables synthetic benchmarking using simulated postings or new test sets, expanding the system's capacity for ongoing validation as models, taxonomies, and labor market data evolve.

In sum, the Evaluation Component ensures trustworthy monitoring of system performance across tasks such as translation, deduplication, and metadata extraction. It provides both quantitative metrics and qualitative scoring, supporting the iterative improvement of AI components and enhancing the overall robustness of MicroIDEA's labor market intelligence pipeline.

F. System Architecture

F.1 Operational Architecture: Cloud and Local Components

The MicroIDEA system is built on a modular architecture designed to balance performance, flexibility, and cross-border scalability. It operates on a hybrid infrastructure that strategically separates components between a centralized cloud server and a dedicated on-premise AI server—ensuring that each subsystem runs in an environment optimized for its specific function (Figure 3).

Data acquisition workflows, including job crawling, preprocessing, and initial translation, are executed on a cloud-based server. This environment handles the ingestion of job postings from multiple national portals, applies cleaning routines, and stores both raw and processed data in a centralized relational database - **MariaDB**. The cloud setup also supports remote access and collaboration, and serves as the backend for the web-based visualization layer powered by **Microsoft Power BI**, which enables rich, interactive dashboards and reporting.

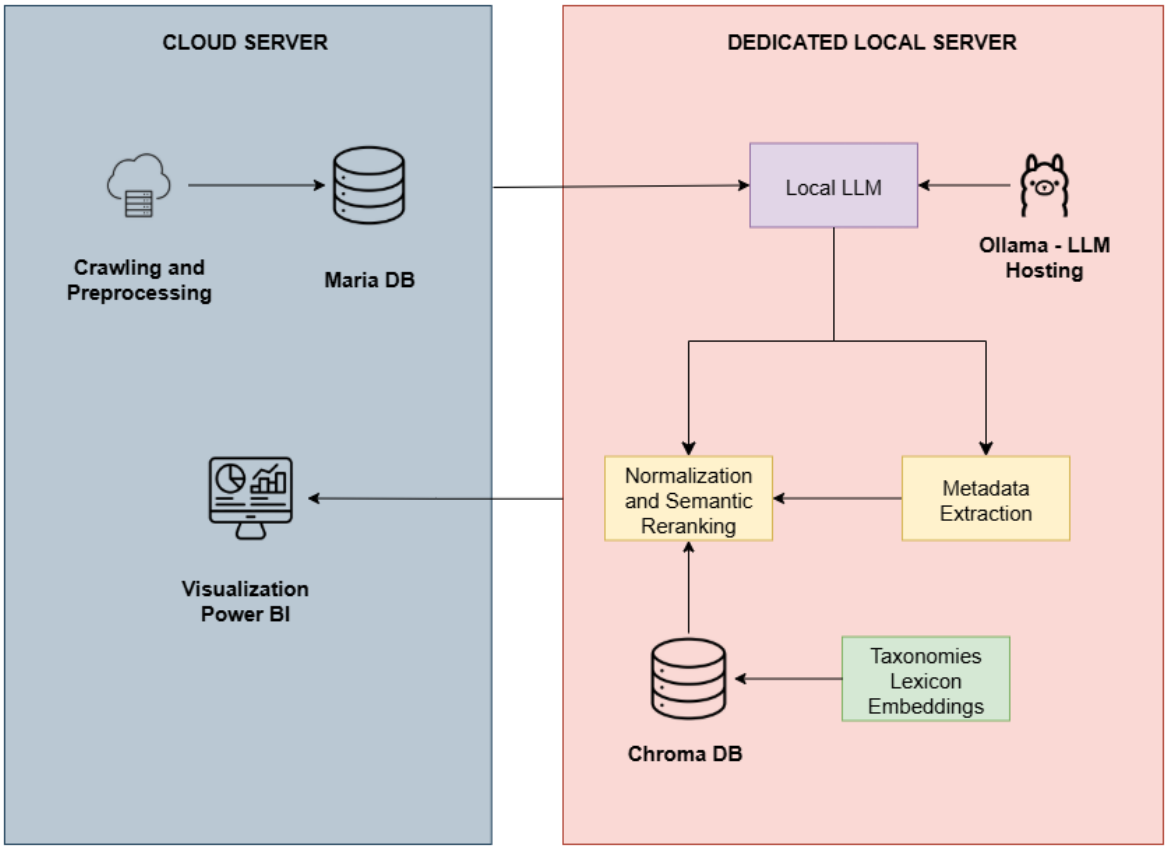

Figure 3: System Architecture

For computationally intensive tasks such as translation, deduplication validation, metadata fields extraction and normalization, and semantic reranking, MicroIDEA leverages a dedicated local server equipped with an **NVIDIA RTX 4090 GPU**. This server runs the **Ollama framework** to serve open-source large language models (e.g., Phi-4, LLaMA-3) locally, enabling privacy-preserving and low-latency inference without dependence on third-party APIs. When broader model capabilities are needed, the system can dynamically offload tasks to Azure-hosted commercial LLMs.

To support semantic similarity tasks—such as skill and occupation normalization—the system uses dense text embeddings generated by transformer models. These are indexed using **ChromaDB**, a lightweight vector database optimized for fast and scalable similarity search.

Enriched metadata produced through AI-driven processing is returned to the cloud database, forming the foundation for downstream applications. By decoupling components across cloud and local infrastructures, MicroIDEA achieves high modularity, reproducibility, and continuous processing capabilities—offering a robust foundation for real-time, privacy-conscious labor market intelligence at scale.

F.2 Implementation

The following subsections describe how the methodological components are instantiated as a modular software architecture, following a two-phase data pipeline (Figure 4). The MicroIDEA system is designed as a two-phase architecture that transforms raw, unstructured job postings from various online sources into structured, semantically enriched, and standardized labor market data. This architecture integrates robust data engineering practices with cutting-edge AI capabilities and follows a clear, traceable flow from data ingestion to final output. Each phase addresses specific technical challenges—ensuring not only high processing accuracy but also adaptability to multilingual and cross-country contexts.

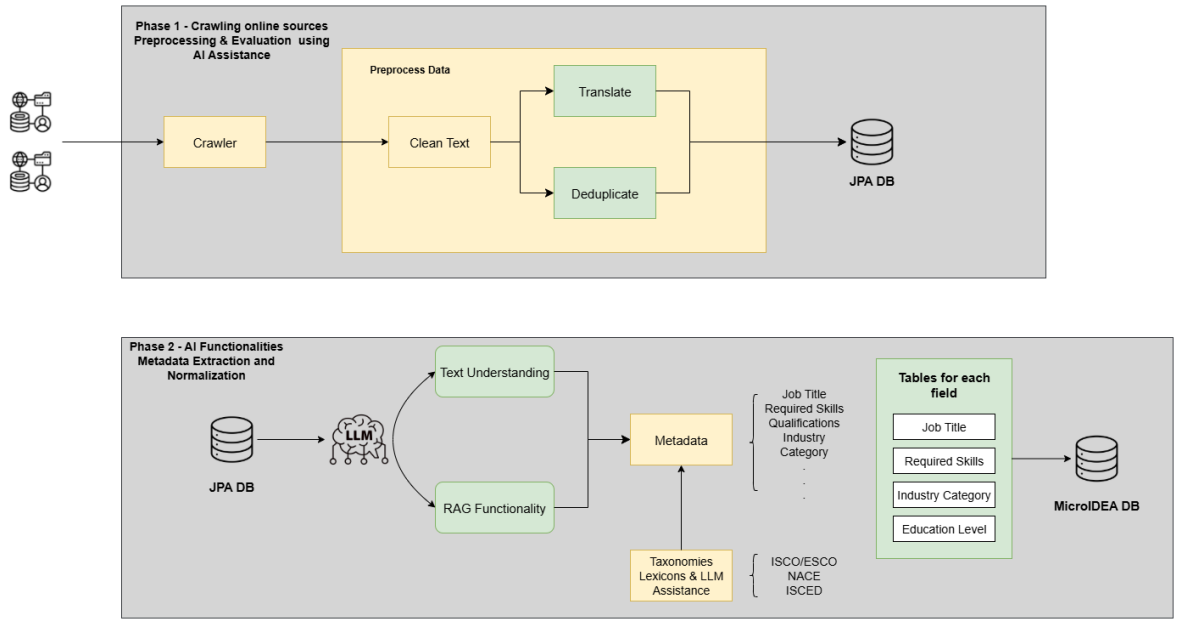

Figure 4: Implementation Phases

The diagram in Figure 4 illustrates the modular architecture of the MicroIDEA pipeline, consisting of two main phases: (1) Crawling and Preprocessing and (2) AI-based Metadata Extraction and Normalization. Each component in this pipeline is purpose-built to address a specific stage in the transformation of raw job data into standardized, multilingual labor market intelligence. The tables below provide an overview of these components, detailing their function, rationale, and the tools or models employed to implement them (Table 1,2).

Component	Purpose	Rationale	Tool/Models Used
Crawling	Collect raw job postings from web portals	Provides fresh, geographically diverse job ads from Greece, Cyprus, Spain	BeautifulSoup[5], MariaDB[6]
Text Cleaning	Normalize and sanitize raw job text	Removes noise (HTML, encoding errors, etc.) to prepare for accurate NLP processing	BeautifulSoup, chardet[7], re[8]
Translation	Convert Greek/Spanish text to English	Standardizes language for LLM prompts and enables consistent schema-based extraction	Argos[9], Phi-4 (LLM)[10]
Deduplication	Identify and flag semantically identical job postings	Eliminates redundancy for accurate labor market analytics	WordLlama[11], Phi-4

Table 1: Phase 1 - Component-Level Description

Component	Purpose	Rationale	Tool/Models Used
Metadata Extraction	Extract structured info such as job title, skills, qualifications, etc.	Converts unstructured text into schema-based structured records	Phi-4, LLaMA-3-8B[12], Pydantic[13]
Normalization - RAG Functionality	Align extracted fields to EU standards (ESCO, NACE, ISCED)	Ensures semantic comparability across countries and languages	Sentence Transformers[14], ChromaDB[15], Phi-4

Table 2: Phase 2 - Component-Level Description

[5] https://www.crummy.com/software/BeautifulSoup/
[6] https://mariadb.org/
[7] https://pypi.org/project/chardet/
[8] https://docs.python.org/3/library/re.html
[9] https://github.com/argosopentech/argos-translate
[10] https://huggingface.co/microsoft/phi-4
[11] https://huggingface.co/dleemiller/word-llama-l2-supercat
[12] https://huggingface.co/meta-llama/Meta-Llama-3-8B
[13] https://docs.pydantic.dev/latest/
[14] https://huggingface.co/sentence-transformers
[15] https://www.trychroma.com/

F.2.1 Phase 1: Data Acquisition and Preprocessing

Phase 1 of the MicroIDEA pipeline addresses the critical challenge of collecting and preparing real-world job postings for semantic enrichment. This stage begins with the automated crawling of diverse online sources, capturing unstructured postings in their raw form, often written in different languages and varying in structure and completeness. Preprocessing then standardizes this data through three core steps: text cleaning (removing noise such as HTML tags and encoding errors), machine translation (converting non-English content into English for consistent downstream processing), and deduplication (identifying and filtering semantically identical entries). This rigorous preparation ensures that the input passed to later AI-based modules is both linguistically normalized and analytically reliable. The example below illustrates how a raw job posting is transformed into a cleaned, translated, and deduplicated record ready for metadata extraction.

Field	Value	Component
id	326	Crawler
source	jobfind	Crawler
title	Σερβιτόρος	Crawler
description	Περιγραφή By the Sea Beach Bar Restaurant Ζητάμε Σερβιτόρο για τη θερινή περίοδο (Ιούνιος - Σεπτέμβριος). Απαραίτητα Προσόντα Ευχάριστη προσωπικότητα και ομαδικό πνεύμα Διάθεση για εργασία και εξυπηρέτηση πελατών Προηγούμενη εμπειρία σε αντίστοιχη θέση (επιθυμητή αλλά όχι απαραίτητη) Καλή γνώση αγγλικών Απόφοιτος Λυκείου Παροχές Ευχάριστο περιβάλλον εργασίας δίπλα στη θάλασσα Ανταγωνιστικές αποδοχές Δυνατότητα διαμονής (κατόπιν συνεννόησης) Διατροφή	Crawler
company	BY THE SEA LUXURY SUITES	Crawler
location	ΘΑΣΟΣ	Crawler
scraped	2025-03-13 20:02:43	Crawler
posted	2025-03-13	Crawler
type	Πλήρης απασχόληση	Crawler
url	https://www.jobfind.gr/JobAd/View/GR/Theseis_Ergasias/eae5847d-47fb-4ad7-9dc3-1ab298245fea/	Crawler
en_title	Waiter	Translation
en_description	Description By the Sea Beach Bar Restaurant We ask for waiter for the summer season (June - September). Essential Qualifications Pleasant personality and team spirit Disposal for work and	Translation

	customer service Previous experience in a corresponding position (desirable but not necessary) Good knowledge of English High School graduate Facilities Pleasant working environment next to the sea Competitive earnings Accommodation possibility (upon request) Nutrition	
Is duplicate	0	Deduplication

Table 3: Phase 1 Example from Greek Portal

Data Collection

Job postings are retrieved through automated crawling of public employment portals across Greece, Cyprus, and Spain. Designed for robustness and scalability, the crawler operates autonomously and on a fixed time schedule, scraping content from selected job portals using country-specific configurations.

The crawling process begins with the system sending HTTP GET requests to retrieve the HTML content from these portals. Adherence to ethical web scraping guidelines is ensured by respecting each site's rules, which govern the accessibility of site data for such purposes. Using Python's BeautifulSoup library, the system parses the HTML content to extract data from predefined HTML elements, identified by tags and attributes, corresponding to the necessary job posting information.

It is executed **once per day**, using **country-specific time slots** to optimize network and compute resource usage while avoiding overloading source servers. Each run targets only the postings that were **published the day before**. This approach ensures freshness while providing a stable temporal anchor for time-based analysis (e.g., daily trends). The crawler keeps a log of the last execution time per portal and uses posting date filters to prevent duplicate records. Execution is managed through a centralized job scheduler. Completion logs and error reports are stored for monitoring, and retry logic is implemented with exponential backoff in the case of transient failures or portal downtime.

To accommodate the differences in structure, layout, and update frequency of each portal, the crawler is customized for each source. It extracts fields such as job title, location, company name, job description, posting date, etc (Table 3). A full schema of raw data fields including type and description is provided in **Appendix A**. The crawled data is then stored in the JPA database, with separate schemas and staging tables per country to ensure clean data separation and traceability.

Text Cleaning

Following data collection, we proceed to the preprocessing phase, which is particularly critical for text-based fields such as job descriptions. Given that raw text often contains irrelevant or noisy elements, we apply a series of normalization techniques, including encoding correction, removal of HTML tags and URLs, elimination of special characters and standardizing lowercase. These steps ensure that data are clean and consistent for further analysis (Table 4).

Preprocessing Step	Description
Removing HTML tags	BeautifulSoup is used to parse HTML content and remove all HTML tags, retaining only the plain text
Unicode normalization	Fixing encoding problems involves identifying and correcting these issues to ensure that the data can be properly encoded in UTF-8
Remove URLs	Regular expressions are used to eliminate hyperlinks that add noise without semantic value.
Removing special characters	Removes emojis and non-alphanumeric symbols, preserving basic punctuation (e.g., . , ; : ! ? ’ ” ( ) / - @ €), using regular expressions.

Table 4: Text Cleaning Steps

Translation

As previously detailed in the methodology section, all non-English job titles and descriptions are translated into English to ensure consistent representation across the pipeline. This step enables effective integration with downstream LLM-based components, which rely on English input for accurate metadata extraction and semantic normalization.

The translation module initially relied on Argos Translate [17], an open-source offline library integrated into the scraping tools for Greece, Cyprus, and Spain. Although lightweight and easy to deploy, Argos frequently misinterpreted domain-specific terminology—particularly in job descriptions—resulting in semantic inconsistencies that negatively affected metadata accuracy.

These issues were identified through the evaluation component outlined in the methodology section, which assessed translation outputs against curated ground truth data using the LLM-as-a-Judge framework. Based on these findings, the system was upgraded to use Phi-4 [18], a locally hosted large language model that demonstrated significantly improved contextual understanding and domain sensitivity. To ensure consistent and reproducible outputs, Phi-4 operates with a low temperature setting (0.2). The translation process applies two tailored prompt strategies depending on the input type, distinguishing between job titles and longer descriptive fields.

- **Job Titles**: A strict prompt ensures literal translation and preserves formatting and special characters (Figure 4).
- **Job Descriptions**: Longer texts are segmented into semantically coherent chunks to retain contextual integrity, especially in lists or structured blocks (Figure 5).

```
model = 'phi-4'
temperature = 0.2
system_prompt = """
You are a professional translator. Your ONLY task is to translate the text to
English.
RULES:
1. Translate the input EXACTLY without adding ANY extra words.
2. DO NOT explain, clarify, or comment on the translation.
3. DO NOT infer missing details (e.g., locations, names).
4. If the text is already in English, keep the same.
5. If the input is empty, return an empty string.
6. Preserve special characters (hyphens, slashes, etc.).
"""
user_prompt = f"""
Translate the following text to English. Return ONLY the translated text, NOTHING
ELSE.
Text: “{text}”
Translation:
"""
```

Figure 5: Prompt for Job Title Translation

```
model = 'phi-4'
temperature = 0.2
system_prompt = """
You are a professional translator. Translate the text to English.
RULES:
1. Output ONLY the translated text.
2. Preserve numbers and special characters.
3. Never add explanations.
"""
user_prompt = f"Translate this to English:\n{chunk}"
```

Figure 6: Prompt for Job Description Translation

Both the original and translated versions are stored in the JPA database (`title`, `en_title`, `description`, `en_description`), supporting transparency, traceability, and bilingual auditability.

This LLM-based translation approach significantly improves semantic fidelity and ensures that specialized terms are retained, resulting in better performance across all subsequent components in the MicroIDEA pipeline.

Deduplication

The deduplication process is applied to job postings after text cleaning and translation are completed. It operates in two phases: a preliminary filtering stage using heuristic-based rules and a more refined semantic evaluation phase powered by LLMs (Figure 7).

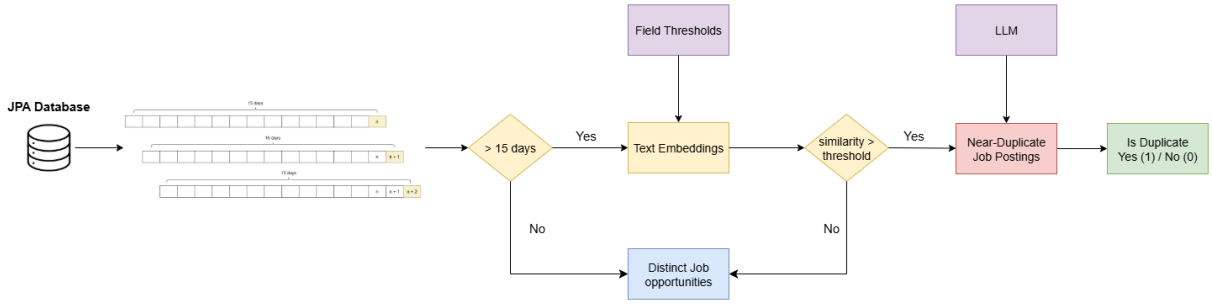

Figure 7: Implementation diagram of the deduplication process

First, a dedicated process has been developed to retrieve recent job postings from the JPA database and perform comparative analysis against previously stored entries. Through exploratory analysis of job posting datasets and discussions with domain experts, we observed that a large

portion of these near-identical ads, often generated by bots or automated systems, appear on a daily or weekly basis. These are likely meant to keep the listing visible at the top of the job boards. To reduce the impact of such spam-like activity, we adopt a rolling 15-day sliding window when determining whether two job advertisements should be considered as duplicates (Figure 8). A sliding window is a temporal filtering technique used to compare new data against a recent subset of past entries within a defined time frame. In this context, that means that each newly retrieved job posting is checked for duplication only against ads published in the previous 15 days. This approach balances recency and relevance, allowing the system to effectively detect reposted or slightly altered duplicates while maintaining computational efficiency and avoiding unnecessary comparisons with outdated records.

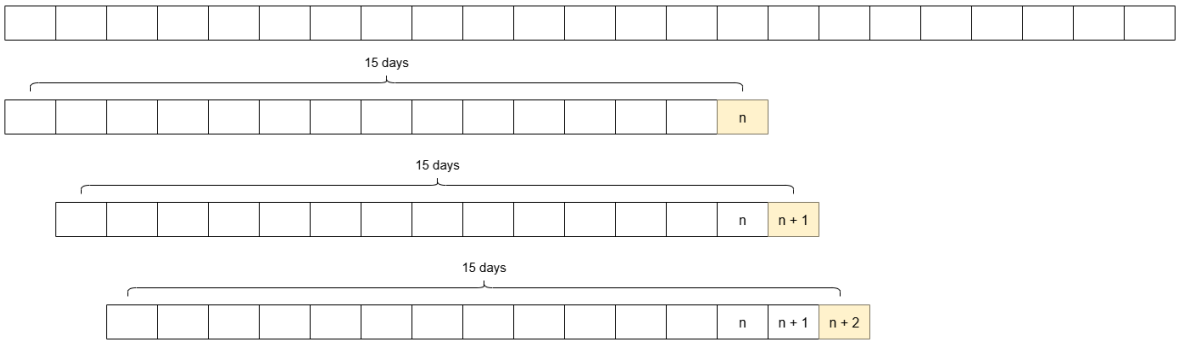

Figure 8: Rolling 15-Day Sliding Window for Duplicate Detection

Next, for the job postings that fall in the window, we employ textual embeddings to detect near-duplicate job postings, utilizing efficient word representations derived from LLMs. This methodology is implemented through WordLlama, a lightweight NLP toolkit optimized for fuzzy deduplication, semantic similarity, and ranking tasks [19]. To assess similarity between job postings, we compare multiple fields—`en_description`, `en_title`, `company`, and `location`—each contributing differently to the deduplication process. To accommodate these differences, we adopt a tiered similarity threshold strategy, tailored to the semantic characteristics and variability of each field, as summarized in the table below (Table 5).

Field	Similarity Threshold	Rationale
Company/Location	0.9	High-precision matching to minimize typographical errors
en_title	0.8	Accommodates phrasing variability while preserving semantic equivalence [20]
en_description	0.7	Lenient threshold for unstructured text, accounting for verbosity and structural differences [11]

Table 5: Field-specific Similarity Thresholds

If the above criteria are met, a pair of job postings is considered as **near duplicate**. While efficient, this method struggled with semantically similar but lexically different descriptions, often failing to detect near duplicates or falsely grouping distinct postings under the same category.

Therefore, based on the limitations observed in the heuristic approach, a second version of the deduplication module was developed following internal evaluation. This phase introduced semantic validation using local LLMs and specifically models such as Phi-4 and llama-3-8b [21].

In this enhanced version, candidate duplicate pairs identified via heuristics are passed to the LLM for semantic comparison. The model receives both job descriptions, the title, company name, and location for each entry, and is prompted to assess whether the two postings represent the same job opportunity. The prompt is designed to focus the model's attention on key fields—`en_title`, `location`, `company`, and `en_description`—and to guide its decision-making by instructing it to ignore superficial differences such as company naming variations (e.g., parent vs. subsidiary) or slight location changes. The model is explicitly instructed to focus on **semantic equivalence** rather than surface-level wording or formatting (Figure 9). The output includes a binary label (`is_duplicate`) and a justification statement, which is stored in the database for transparency and auditability. Duplicates are assigned a shared `duplicate_group_id`, while distinct entries are retained as unique records.

```
model = 'phi-4'
temperature = 0.2
prompt = (
"<s>[INST] You are comparing two job postings to decide: if they refer to the same
job.
    "Pay attention to the following fields: "
    "en_title, location, company, and en_description. "
    "Ignore differences for company name variations and "
    "parent/subsidiary variations. "
    "Ignore location slight variations. "
    "Your task is to judge **semantic equivalence** – "
    "ignore minor wording or formatting differences.\n\n"
    "Respond ONLY in the following JSON format:\n\n"
    "{\n"
    "  \"isDuplicate\": \"Yes\" or \"No\",\n"
    "  \"howDifferent\": \"Explain why they are different, "
    "or say 'Minor variations only' if they are the same\",\n"
    "  \"jobDescDifference\": \"Explain any differences in the job description\",\n"
    "  \"locationDifference\": \"Explain any differences in location\",\n"
    "  \"companyDifference\": \"Explain any differences in company name\"\n"
    "}\n\n"
    "Do not include anything outside the JSON object.\n\n"
    + html_content +
    " [/INST]"
)
```

Figure 9: Prompt for Near Duplicate Pairs Evaluation

An example of such LLM-based judgments is presented in Table 6, which shows near-duplicate pairs detected from the first stage, but labeled in the second stage as non-duplicate pairs with accompanying explanations. These examples demonstrate the system's ability to distinguish subtle but important differences in title and description —even when the postings appear similar at a glance. By combining similarity metrics with contextual LLM evaluation, this component ensures high-quality deduplication across multilingual and domain-diverse job data.

Field	Job Posting #1	Job Posting #2	Is Duplicate	LLM Justification
Title	**Salesman/seller** in a bakery	**Seller / saleswoman** in a bakery	No	The highlighted text indicates minor text difference, but the difference in the employment
Location	ΚΕΝΤΡΟ ΘΕΣΣΑΛΟΝΙΚΗΣ	ΚΕΝΤΡΟ ΘΕΣΣΑΛΟΝΙΚΗΣ		
Company	ΖΑΧΑΡΟΠΛΑΣΤΕΙΟ	ΖΑΧΑΡΟΠΛΑΣΤΕΙΟ		

Description	Description for Sale/Sales from a bakery in the center of Thessaloniki. **Part-time job (five-hour afternoon)**. Required qualifications, desired experience in a corresponding position, **disposal** for work, consistency and professionalism. Good Knowledge of English and good communication skills	Description for Sale/Sales from a bakery in the center of Thessaloniki. **full-time job (five-day eight-hour work)**. Required qualifications, desired experience in a similar position, **availability** for work, consistency and professionalism. Good Knowledge of English and good communication skills		type (Part time in job position 1 / Full time in job position 2) indicate semantic difference and represent unique job opportunities.

Table 6: Example of Near-Duplicate Pair Evaluation

F.2.2 Phase 2: AI-Driven Extraction and Semantic Enrichment

In Phase 2, preprocessed job data is processed by core AI modules built around LLMs and RAG techniques. This phase performs two main functions: extracting structured metadata from unstructured text and aligning this information with standardized taxonomies.

The pipeline begins with **text understanding**, where LLMs are used to identify and extract fields such as job titles, required skills, qualifications, industries, and education levels directly from the job descriptions (`en_description`) (Figure 10).

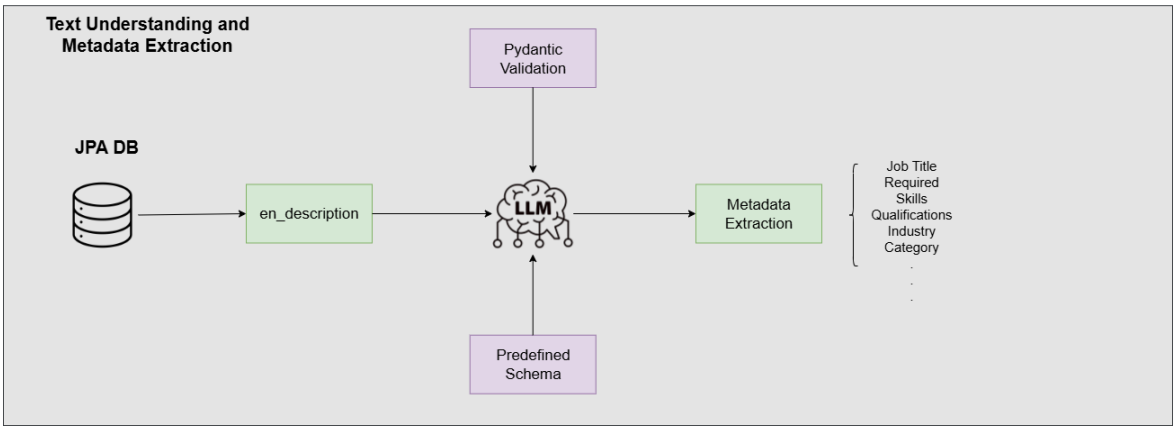

Figure 10: Text Understanding and Metadata Extraction

Based on the job posting example presented in Table 3, Table 7 illustrates the output of the structured metadata fields extracted from the unstructured text using AI-assisted methods. Each

field corresponds to a specific element in the metadata schema, representing a semantically enriched interpretation of the original posting.

Field	Value
Advertising company	By The Sea Beach Bar Restaurant
Hiring Company	By The Sea Beach Bar Restaurant
location area	
Job title	Waiter/Waitress
occupation	Hospitality Staff
Job season	Summer (June - September)
Job responsibilities	[]
Required skills	[‘Customer Service’, ‘Work in a team environment’, ‘Good Knowledge of English, ‘Pleasant Personality’]
qualifications	[‘Previous experience in a corresponding position desirable but not necessary’]
contact	
salary	Competitive earnings
Education level	High School graduate
Employment type	
Job benefits	[‘Accomodation possibility’]
Job positions	1
Working hours	
Working experience	Previous experience in a corresponding position desirable but not necessary
Industry category	Hospitality and Tourism
languages	[‘English’]
Remote work	On-site
Company description	A beach bar restaurant near the sea.
Work environment	Pleasant working environment next to the sea

Travel requirements	
Provided equipment	

Table 7: Example of Structure Metadata Fields

Next, the **RAG functionality** enriches this extraction process by incorporating external knowledge—drawing from curated taxonomies (ISCO, ESCO, NACE, ISCED)—to resolve ambiguities, standardize terminology, and ensure alignment with cross-country classification systems (Figure 11).

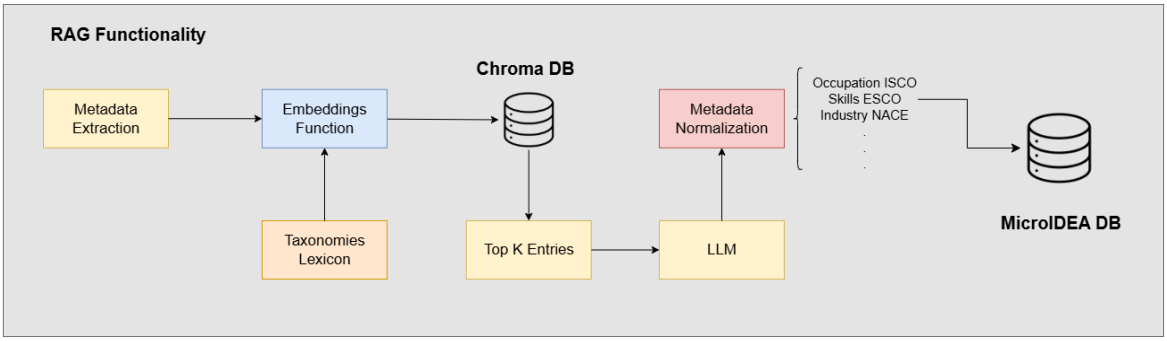

Figure 11: RAG Functionality for Field Normalization

Metadata fields like `required skills`, `job title`, `industry category`, `education level` are the input for the RAG process and the output of this process is a collection of structured field-specific tables (e.g., Job Title, Skills), each representing a clean and harmonized view of the job posting's semantic content. Applying this process to the above example, we have the following tables with corresponding Information (Table 8-11)

Required skill	Skill esco	Is new skill	Skill type	reuselevel
Customer service	customer service	0	knowledge	sector-specific
Work in a team environment	Work in teams	0	skill/competence	transversal
Good knowledge of English	English	0	knowledge	transversal
Pleasant personality	Demonstrate good manners	0	skill/competence	transversal

Table 8: Example of Skills Normalization

Job title	Occupation Title	ESCO code	ISCO code
Waiter/Waitres	waiter/waitress	51312	5131

s			

Table 9: Example of Job Title Normalization

Industry Category	Class Id	Industry Nace
Hospitality and Tourism	79.9	Other reservation service and related activities

Table 10: Example of Industry Normalization

Education level	ISCED Level
High School graduate	3

Table 11: Example of Education Level Normalization

Text Understanding and Metadata Extraction

The first implementation of the metadata extraction process in MicroIDEA was designed as a proof-of-concept to validate the effectiveness of schema-guided (see Figure 2) LLM inference for structured data extraction from unstructured job descriptions. This phase prioritized extraction completeness and adherence to a well-defined schema, while laying the groundwork for later optimizations in speed, token efficiency, and disambiguation accuracy.

The core idea was to prompt an LLM to extract predefined fields using a structured schema based on a Pydantic[16] class, which not only described the expected structure but also enforced post-extraction validation. Each field in the schema was carefully defined, drawing from both domain knowledge and best practices established by prior work [22], guided us in choosing the right definitions to extract the information we want from job advertisements.

A full schema of extracted fields is provided in **Appendix B - Table B1**. To improve readability and maintain structure, the schema is grouped into the following categories:

- **General Job Information**: job title, occupation, location, job season, employment type
- **Job Requirements**: required skills, qualifications, education level, languages, working experience
- **Employer and Contact**: hiring company, advertising company, company description, communication contact
- **Compensation and Logistics**: salary details, working hours, benefits, travel requirements, remote option, provided equipment, number of positions
- **Job Description Insights**: job responsibilities, work environment details

This design enabled precise mapping between free-text job postings and structured relational fields. The extraction process was carried out using LLMs via a dynamically generated prompt that incorporated the full schema inline (Figure 12). The user prompt instructed the model to extract data strictly in the prescribed JSON format. The system prompt defined the model's role as an extractor and set strict generation constraints—such as avoiding hallucinations and returning empty values for unspecified fields.

[16] https://python.useinstructor.com/

```
model = "meta-llama-3.1-8b-instruct"
temperature = 0.2
system_prompt = """You are a helpful assistant that outputs JSON data for job
posting comparisons."""
user_prompt = """<s>[INST] Please extract the job posting details from the following
text and format the response strictly according to the schema provided below.
Return only a JSON object in this structure, without any additional text or
explanations: {schema_prompt}
Strictly return only a JSON object in this structure with accurate data extracted
from the job description text. {description} [/INST]"""
```

Figure 12: User and System Prompt

According to the ILO[17] report, ISCED 2011[18] and ISCED-F 2013[19] frameworks, qualifications are not officially recognized attestations of learning outcomes. These can be acquired through formal education or validated through informal means. Skills, on the other hand, are defined as the innate or acquired ability to apply knowledge in order to perform tasks and responsibilities associated with a specific job. Skills can be categorized into several types: job-specific or technical skills, which are unique to a particular occupation and include specialized knowledge, familiarity with tools or machinery, and understanding of specific materials or products; basic skills, such as literacy, numeracy, and ICT, which are essential for further training and the acquisition of more complex skills; and transferable skills, which are applicable across various jobs and industries and include cognitive abilities, physical dexterity, communication skills, and socio-emotional or behavioral competencies.

In the job advertisements we analyze, the distinction between skills and qualifications is often unclear, which necessitates further conceptual clarification. Certain competencies may function as both a skill and a qualification—for instance, a driving license is a formal qualification, while the ability to drive is the corresponding skill. Conversely, a requirement such as "five years of experience" represents a qualification but not necessarily a skill.

To address this ambiguity, it is essential to place particular emphasis on prompt engineering tailored to these two categories, providing as clear and distinct definitions as possible. At this stage of implementation, we are systematically reviewing and reassigning all listed skills and qualifications to their appropriate categories, using consistent model settings and the prompt instruction (Figure 13)

```
model = "meta-llama-3.1-8b-instruct"
temperature = 0.2
prompt = (
    "You are an AI assistant tasked with classifying items "
    "from job descriptions.\n\n"
    "Definitions:\n"
    "- Skills: Abilities, proficiencies, or competencies required to perform a
task.\n"
    "- Qualifications: Previous experience, degrees, certifications, or formal
requirements.\n\n"
    "Classify the following text as either 'Skill' or 'Qualification':\n\n"
    f"Item: '{item}'\n\n"
    "Response should only be 'Skill' or 'Qualification'."
)
```

Figure 13: Prompt for Skills and Qualifications Refinement

[17] https://www.ilo.org/

[18] https://uis.unesco.org/sites/default/files/documents/international-standard-classification-of-education-isced-2011-en.pdf

[19] https://uis.unesco.org/sites/default/files/documents/international-standard-classification-of-education-fields-of-education-and-training-2013-detailed-field-descriptions-2015-en.pdf

This refinement enhances data usability for filtering and analysis. Finally, the metadata fields are stored in the corresponding table in the MicroIDEA Database.

Following the initial implementation, after extended evaluation trials a new approach was designed to address key limitations uncovered through evaluation: inefficiencies in processing time, redundancy from postprocessing steps, and ambiguity in distinguishing between semantically close but conceptually different fields—especially "skills" and "qualifications". The goal of this phase was to consolidate metadata extraction into a single, more efficient step while enhancing extraction precision. This was accomplished through improved prompt engineering, schema refinement, and targeted model configuration (Table 12).

Aspect	Version 1	Version 2
Model	Llama-3-8b	Phi-4
Temperature	0.2	0.1
Prompt Engineering	Basic Schema	Schema + exclusion/inclusion rules + examples
Skill/Qualification Distinction	Post processing refinement	Integrated at extraction

Table 12: Clarification of Changes in Metadata Extraction

In the version 2, most of the fields of version 1 are retained, but with improved descriptions, stricter definitions, and revised examples. Notably, fields such as required_skills, qualifications, and required_languages were refined to reflect clearer semantic boundaries. For instance, a driver's license is now consistently captured as a qualification, while "driving experience" is identified as a skill. A full listing of the updated schema used in this phase is provided in **Appendix B, Table B2**. The prompt structure remained similar to Phase 1 in format but was enriched with these decision boundaries. Below is a simplified representation of the configuration (Figure 10).

```
model = "phi-4"
temperature = 0.1
system_prompt = """You are a helpful assistant that outputs JSON data for job
posting comparisons. Please also note:
- Exclude as non-skills: "ability to travel", "ability to work in shifts", "passion
for technology", "interest for cooking", "talent in communication", "flexibility for
shift work", "willingness to", "appetite for work", "disposal for work", "possession
of a bicycle".
- Include as skills: "experience in cooking", "experience in driving"."""
user_prompt = """<s>[INST] Please extract the job posting details from the following
text and format the response strictly according to the schema provided below.
Return only a JSON object in this structure, without any additional text or
explanations: {schema_prompt}
Strictly return only a JSON object in this structure with accurate data extracted
from the job description text. {description} [/INST]"""
```

Figure 14: Model Settings for the Version 2 of Metadata extraction

The results of this approach showed substantial improvement in classification precision, particularly in fields that were most error-prone in version 1. More importantly, it enabled direct ingestion of LLM outputs into the downstream normalization pipeline without the need for intermediate correction steps which increase the cost in time and tokens.

RAG Functionality for Metadata Fields Normalization

The Normalization stage in MicroIDEA ensures that extracted metadata—such as skills and occupations—is semantically consistent and aligned with European and international classification systems. This process is essential not only for harmonizing terminology across multilingual job postings but also for enabling accurate visualization, aggregation, and cross-country comparisons in labor market analytics. Specifically, normalization maps extracted entities to official European taxonomies such as ESCO (skills and occupations), NACE (industry classifications), and ISCED (education levels).

The ESCO taxonomy provides a multilingual classification of occupations, skills, and qualifications. In the context of MicroIDEA, it is primarily used to normalize skills and occupations extracted from job postings. For skills, the normalization is structured to align with ESCO's skills pillar (Table 13), which

Field	Description
Skill ESCO	column maps the raw skill to its corresponding ESCO preferred label as a broader skill
Skill type	ESCO distinguishes between skill/competence concepts and knowledge concepts
Is new skill	is a boolean variable with value 0, if the preferred label is from ESCO lexicon, or value 1, if the model has created a new one more suitable.
Reuse level	which indicates how widely a knowledge, skill or competence concept can be applied. ESCO distinguishes four levels of skill reusability (Transversal knowledge, Cross-sector knowledge, Sector-specific knowledge)

Table 13: Skills ESCO Taxonomy Mapping

provides a comprehensive list of knowledge, skills and competences relevant to the European labor market. The **ESCO skills pillar** consists of 13939 skills and distinguishes between skill/competence concepts and knowledge concepts by indicating the skill type. Each of these concepts come with one preferred term and a number of non-preferred terms in each of the 28 ESCO languages. ESCO as well provides an explanation (metadata) for each skill profile such as a description, scope note, reusability level and relationships (with other skills and with occupations).

Similar to skills normalization, job titles are aligned to the ESCO occupations pillar. The job titles are normalized by matching them to standardized occupational classifications from a predefined lexicon. This lexicon is the **ESCO occupation pillar** (Table 14) consists of 3039 occupations, which is built on ISCO-08 which serves as its hierarchical structure. ISCO-08 provides the top four levels for the occupations pillar and ESCO occupations are located at level 5 and lower. In ESCO, each occupation is mapped to exactly one ISCO-08 code.

Field	Description
Occupation Title	column maps the raw job title to its corresponding ESCO preferred label as a broader occupation

ESCO Code	A **hierarchical structure**, where ESCO occupation codes are more fine-grained than ISCO-08
ISCO-08 Code	The corresponding **ISCO-08 unit group** code (4-digit)

Table 14: Occupation ESCO/ISCO Taxonomy Mapping

NACE is a four-digit classification providing the framework for collecting and presenting statistical data according to economic activity in a wide variety of European statistics in the economic, social, environmental, and agricultural domains. This taxonomy helps in normalizing industry categories by matching them to standardized classifications from the NACE lexicon, a four-digit European industry classification system. NACE provides the framework for collecting and presenting statistical data across economic activities, supporting a wide range of European statistics in economic, social, environmental, and agricultural domains. The classification is structured hierarchically, with examples including: Section (e.g., Manufacturing, Code C), Division (e.g., Manufacture of food products, Code 10), Group (e.g., Manufacture of dairy products, Code 10.5), and Class (e.g., Manufacture of cheese, Code 10.51) (Table 15).

Nace Level	Nace Title (Example)	Nace Code (Example)
Section	Manufacturing	C
Division	Manufacture of food products	10
Group	Manufacture of dairy products	10.5
Class	Manufacture of cheese	10.51

Table 15: Industry NACE Taxonomy Mapping

We want to match the educational qualifications with ISCED levels. ISCED is designed to serve as a framework to classify educational activities as defined in programmes and the resulting qualifications into internationally agreed categories. The basic concepts and definitions of ISCED are therefore intended to be internationally valid and comprehensive of the full range of education systems (Table 16).

ISCED Level	Education Level	Definition
0	Early Childhood Education	Pre-school education
1	Primary Education	Elementary school - First stage of basic education
2	Lower secondary education	Middle school – Junior high school
3	Upper secondary education	High School Diploma

4	Post-secondary non-tertiary education	Technical diploma - vocational qualifications
5	Short-cycle tertiary education	Associate degree - some higher diplomas
6	Bachelor's or equivalent level	Bachelor's Degree (e.g., BSc, BA, BEng) – Graduate diplomas
7	Master's or equivalent level	Master's Degree (e.g., MSc, MA, MEng) - Postgraduate diplomas and certificates
8	Doctoral or PhD or equivalent level	PhD - Doctorate

Table 16: Education Level ISCED Taxonomy Mapping

A sample of the curated and pre-embedded taxonomy lexicons used during this process is provided in **Appendix C**.

For semantic normalization, MicroIDEA adopts a RAG approach that combines fast similarity-based search with LLM-powered semantic reranking. Each extracted entity—such as a skill, job title, or industry label—is embedded into a high-dimensional vector space using the *all-mpnet-base-v2* model from *SentenceTransformers*. This model captures contextual semantic meaning beyond surface similarity. All entries from taxonomies are pre-embedded using the same model and stored in the ChromaDB vector database. When a new term is encountered, its embedding is compared to the taxonomy index using cosine similarity. If the similarity score exceeds a predefined confidence threshold, the most similar match is accepted. In cases of ambiguity, the top-k most similar entries are retrieved and passed to a local LLM for semantic reranking.

For this reranking step, MicroIDEA uses the *Phi-4* language model with a low temperature setting of *0.1* to ensure near-deterministic and consistent decisions. The model is prompted with the original term, contextual information (when available), and the list of candidate taxonomy entries. It then selects the most contextually appropriate standardized label and may also generate a brief justification. This hybrid process helps disambiguate similar terms—for example, ensuring that "Data Engineer" is correctly mapped rather than being confused with "IT Technician," even when lexical similarity is high.

The output of this process includes the final normalized label, a confidence score, and additional metadata depending on the taxonomy. For skills, this includes the ESCO preferred label (`skill esco`), skill type (`skillType`), a reuse level (`reuseLevel`), and a novelty flag (`is_new_skill`). Occupation fields include the ESCO label and UUID (`occupation esco`, `esco code`) and the corresponding ISCO-08 code (`isco code`). Similar principles apply for industry and education normalization. This process is repeated for every list element in multi-valued fields such as skills or qualifications.

By grounding free-text content in authoritative, structured vocabularies and combining statistical and semantic techniques, this normalization step enables robust, interpretable metadata and supports reliable cross-country labor market insights.

Outputs Stored in MicroIDEA DB

The complete metadata object is stored in the **MicroIDEA Final Database**, which serves as the output layer of the system and the foundation for downstream applications. This database enables

advanced querying, filtering, and visualization, and integrates with skill dashboards, labor market observatories, and policy-driven tools such as micro-credential recommendation engines. The structured data is linguistically harmonized and semantically interoperable, providing robust support for comparative labor market analytics across countries and sectors.

The design of this metadata layer prioritizes semantic precision and field-level granularity. Every stage of the process is validated—ensuring completeness, consistency, and alignment with international standards. The transformation moves beyond surface-level keyword extraction, capturing latent signals embedded in natural language job descriptions and elevating them into structured, reusable knowledge. As a result, the system can detect not only what roles are being posted, but also which skills are emerging, which qualifications are preferred, and how job expectations vary across regions.

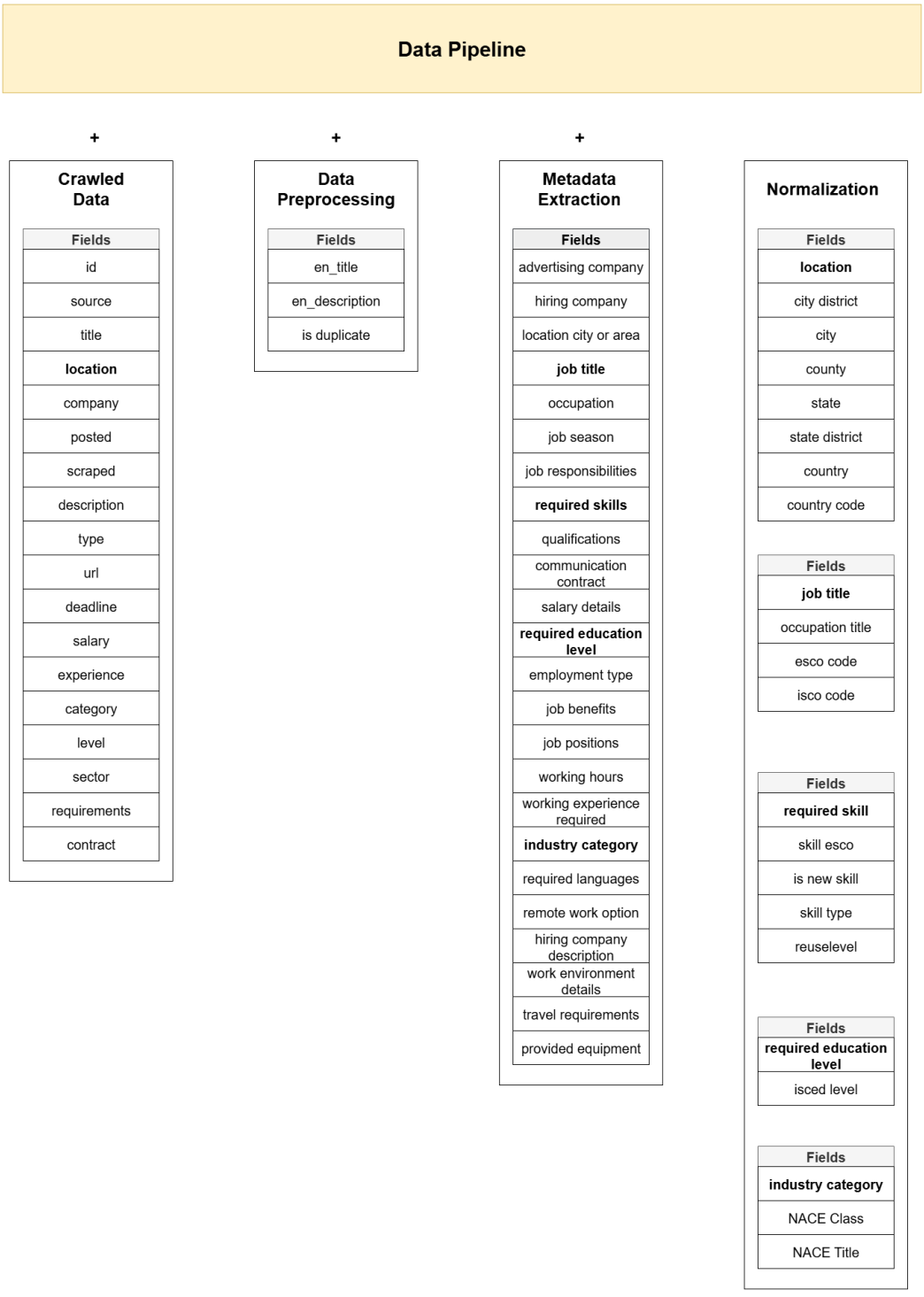

Figure 15: Data Pipeline Across Components

This **field-level enrichment process**—from extraction to normalization—is summarized in Figure 15, which illustrates how raw job content evolves through each stage of the pipeline. The figure highlights the introduction of structured fields, the layering of semantic annotations, and the final generation of standardized records aligned with ESCO, NACE, ISCO, and ISCED taxonomies. The result is a powerful and flexible foundation for labor market intelligence that can inform workforce development, educational policy, and strategic reskilling efforts across Europe.

G. Evaluation

The primary objective of the evaluation process in MicroIDEA is to measure the effectiveness and trustworthiness of each AI component involved in transforming unstructured, multilingual job postings into structured, standardized data. This includes assessing translation quality, deduplication accuracy, and the precision of metadata extraction across various fields such as skills, qualifications, industry codes, and education levels. The evaluation is designed not only to benchmark different model configurations and prompt strategies, but also to ensure that extracted data faithfully represents the source content. By employing both automated metrics and LLM-assisted evaluation strategies, the objective is to maintain high levels of semantic fidelity, reduce hallucinations, and guide iterative improvements across all stages of the system pipeline.

G.1 Test Reference Datasets

To support a robust and representative evaluation of the MicroIDEA system, a curated test-reference dataset was developed encompassing job postings from all three participating countries—Greece, Spain, and Cyprus. This dataset serves as the ground truth for evaluating each LLM-assisted component of the pipeline, including translation, semantic deduplication, metadata extraction, and normalization.

The dataset construction followed a hybrid approach combining automated and human-in-the-loop methods. Initial annotations were generated using GPT-4o, a high-performing language model known for its semantic alignment with human judgments [23]. However, recognizing the limitations of even the most advanced models in capturing nuanced distinctions—particularly in multilingual contexts and domain-specific terminology—each record was manually reviewed and refined by domain experts. This process ensured the reliability and interpretability of the reference data, especially for ambiguous or underspecified job postings.

More specifically, for the **evaluation of the translation component**—covering both Greek-to-English and Spanish-to-English directions—a dedicated test dataset was constructed using a stratified sampling approach. The source data was drawn from a one-month collection period, during which all job postings available in the MicroIDEA database were retrieved. From this full set, a stratified random sampling strategy [24] was applied to extract a 10% evaluation subset, resulting in 300 job postings from Greek portals and 100 from Spanish portals.

Stratification criteria included key metadata fields such as job title and portal of origin, ensuring coverage across a diverse range of occupations and platforms. The sampling distribution was designed to approximate a normal curve, thereby enhancing the representativeness of the dataset while maintaining balance across different job types and data sources. This carefully constructed subset formed the foundation for benchmarking translation quality across multiple systems, allowing for consistent and fair evaluation of semantic fidelity and fluency in multilingual labor market content.

For the **evaluation of the deduplication component**, a dedicated test-reference dataset was constructed comprising 82 job posting pairs labeled as either duplicates or non-duplicates. These pairs were primarily designed to assess the system's ability to distinguish between near-duplicate job ads—a common and challenging scenario in real-world job portals.

Approximately 20% of the dataset was sourced directly from the original job posting corpus collected during the one-month evaluation period. To enhance coverage of edge cases and ensure robustness, the remaining 80% consisted of synthetically generated job postings created using GPT-4o. These synthetic entries were crafted through prompt-based generation, explicitly designed to simulate subtle but semantically significant variations [25].

The prompts used for synthetic augmentation included common sources of ambiguity, such as:

- Variations in job title phrasing (e.g., “Senior Software Engineer (Python)” vs. “Senior Python Developer”)
- Differences in company naming conventions (e.g., “TechCorp” vs. “TechCorp Inc.”)
- Alternative location formats (e.g., “New York, NY” vs. “NYC”)
- Changes in employment type (e.g., full-time vs. contract) or work modality (e.g., on-site vs. remote/hybrid)

Each pair in the dataset is flagged as duplicate or not after human - domain expert annotation. This approach ensures that the test set captures both realistic and edge-case scenarios that are likely to challenge traditional rule-based or surface-similarity deduplication systems.

To **evaluate the metadata extraction component**, a dedicated test dataset of approximately 1,000 job postings was developed using a stratified sampling approach, similarly as in the translation component. The selection criteria included variables such as industry category, job title, and portal of origin, ensuring broad coverage and balance across sectors and data sources.

For each job posting in this dataset, structured metadata was initially extracted using GPT-4o, a state-of-the-art language model known for its strong alignment with human judgment. To ensure maximum reliability and integrity of the ground truth, all model-generated outputs were subsequently reviewed and refined through manual annotation by domain experts. This human-in-the-loop approach was particularly important for resolving ambiguities, interpreting implicit job requirements, and ensuring semantic consistency—especially in multilingual postings where cultural and linguistic nuance may affect interpretation.

This high-quality, validated dataset forms the foundation for benchmarking metadata extraction accuracy across key fields such as job titles, required skills, qualifications, education levels, and industry classifications.

G.2 Evaluation Approach and Metrics

The evaluation of each AI-driven component in MicroIDEA—such as translation, deduplication, and metadata extraction—relies on a combination of quantitative and qualitative assessment methods. Central to our approach is the **reference-based evaluation methodology**, where system outputs are compared against a curated ground truth dataset, enabling consistent and reproducible benchmarking.

To capture the nuanced and often context-dependent nature of job posting interpretation, we adopt the **LLM-as-a-Judge** framework. In this setup, a high-performing language model (e.g., GPT-4o) is used not only to generate outputs but also to evaluate them. This allows for more contextualized and semantically aware judgments, particularly in complex or multilingual scenarios where traditional rule-based evaluation may fall short.

In parallel, we employ standard classification metrics—including **precision, recall, F1-score, and accuracy** [26]—to quantitatively measure system performance across tasks such as metadata extraction and deduplication. These metrics provide a clear view of model effectiveness in terms of correctness, completeness, and reliability.

Together, this dual approach—reference-based benchmarking with both classification metrics and LLM-assisted evaluation—ensures a rigorous and multidimensional understanding of system performance, supporting continuous improvement of the MicroIDEA pipeline.

G.3 Component Evaluation

Translation

This evaluation analyzes the performance of various machine translation systems by comparing their outputs against a reference translation, which serves as the benchmark. The focus of the evaluation is on two key metadata fields: `en_title` and `en_description`. The analysis encompasses both open-source LLMs and a conventional machine translation system, evaluated across two language pairs: Greek-to-English and Spanish-to-English.

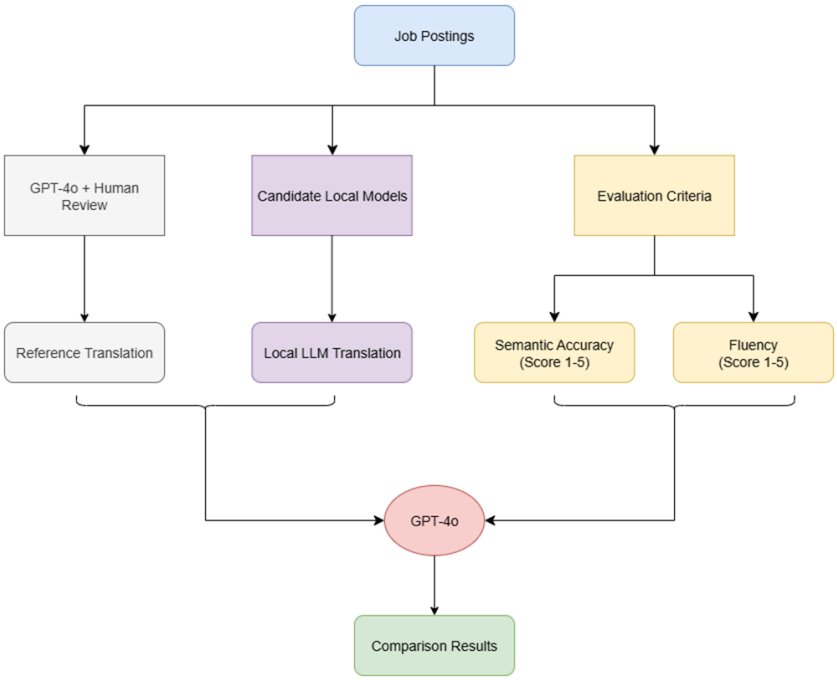

Figure 16: Translation Evaluation Process

The evaluation framework is composed of three main components (Figure 16). The first is the construction of the reference dataset, which consists of a randomly selected sample of job postings. This sample is uniformly distributed across different job portals and job title to ensure broad and representative coverage of the labor market data.

The second component involves the selection of candidate systems. These include three LLM-based models, *deepseek-r1-distill-qwen-14b (DeepSeek-R1)* [27], *mistral-7b-instruct-v0.3 (Mistral 7B)* [28], and *Phi-4*—as well as a traditional, offline machine translation tool, *Argos Translate*. This mix of systems enables a comparative analysis between modern neural approaches and conventional rule-based translation methods.

The third component centers on the evaluation criteria, which rely on two primary metrics: **Semantic Accuracy and Fluency, each rated on a 1 to 5 scale**. Semantic Accuracy assesses the extent to which the candidate translation preserves the intended meaning of the reference, including the correct rendering of key concepts, domain-specific terminology, and contextual nuance. Fluency evaluates the grammaticality and naturalness of the output in the target language, with attention to idiomatic expressions, syntactic structure, and overall readability.

The evaluation revealed several notable trends. **Phi-4** consistently outperformed the other systems across both language pairs, achieving the highest overall scores—**4.21** for Greek-to-English and **4.46** for Spanish-to-English translations. While **Argos Translate** offered faster processing times, this came at the expense of significantly lower translation quality. Notably, all systems performed better on Spanish-to-English translations, suggesting that translation quality is influenced by the characteristics of the source language and domain-specific content.

In conclusion, **Phi-4** was selected as the preferred model for the MicroIDEA pipeline due to its superior balance of semantic fidelity and linguistic fluency across both evaluated language directions. Detailed evaluation scores and performance comparisons for all systems are presented in Appendix D, Tables 1–4.

Deduplication

Evaluating the deduplication component required careful consideration of semantic similarity across near-duplicate job postings—cases where minor textual differences may or may not reflect distinct job opportunities. To benchmark model performance, we used a manually annotated reference dataset of 82 job posting pairs, labeled as either duplicates or non-duplicates. This gold-standard dataset included both real-world samples and synthetically generated edge cases, allowing us to test each model's ability to handle subtle linguistic and structural variations.

The evaluation pipeline, illustrated in Figure 17, was designed around a reference-based comparison framework. Candidate models were prompted uniformly and evaluated using standard classification metrics, with a particular focus on **precision, recall, and F1-score**—the latter serving as the primary metric for balancing false positives and false negatives.

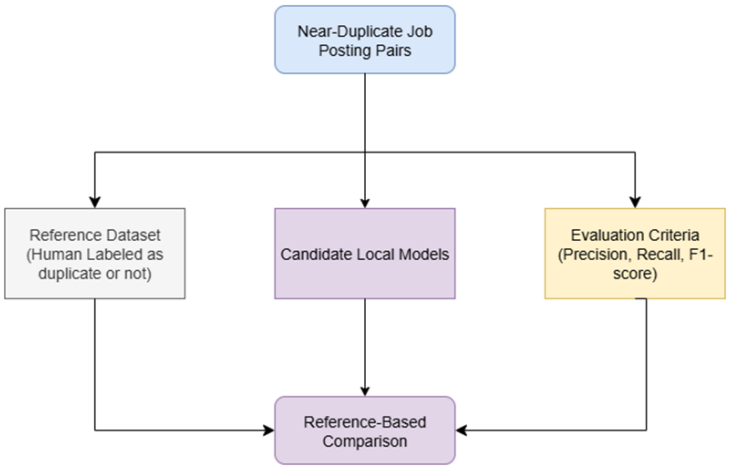

Figure 17. Evaluation pipeline for the deduplication component

All the models performed quite well, but among the four evaluated models, Phi-4 and DeepSeek-R1 emerged as the top performers, each achieving an F1-score of 0.96, indicating exceptional semantic precision and reliability.

Although **Phi-4** and **DeepSeek-R1** performed equally well in terms of accuracy, **Phi-4 was selected as the preferred model** for deduplication in the MicroIDEA pipeline. This decision was based on a strategic alignment with the translation component, where Phi-4 was also chosen as the top-performing model. Maintaining model consistency across components enhances system coherence, simplifies integration, and reduces variability in downstream processing. Detailed evaluation results and metrics are presented in Appendix E.

Metadata Extraction

The evaluation of the metadata extraction component focused on assessing the performance of four leading large language models—**Phi-4**, **Qwen-30B** [29], **DeepSeek-R1**, and **LLaMA-3-8B**—across structured fields derived from job postings collected in **Greece, Cyprus, and Spain**. The goal

was to identify the most accurate and efficient model for extracting key job-related metadata such as job titles, skills, qualifications, industry categories, and education levels in a multilingual setting.

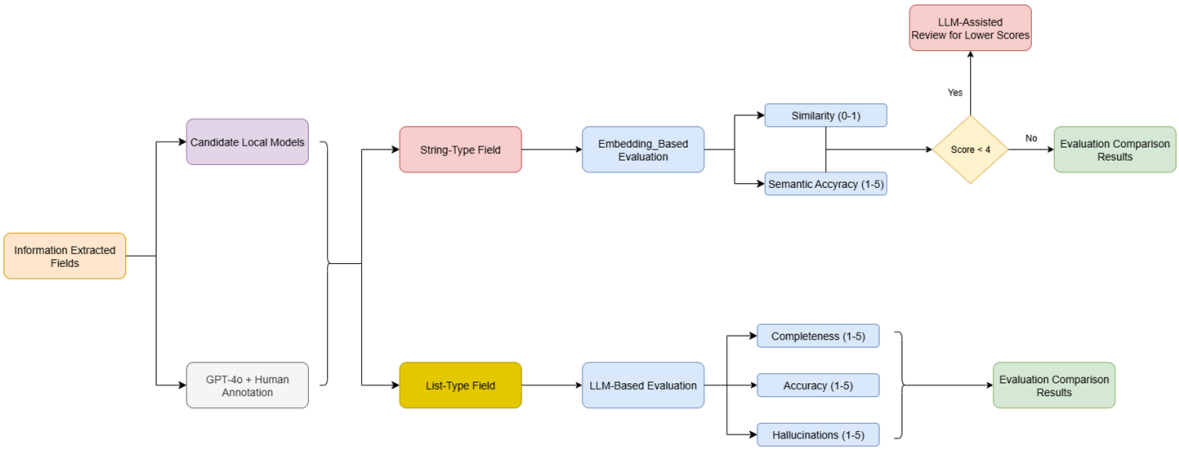

Figure 18. Evaluation pipeline for the Metadata Extraction component

The evaluation pipeline, illustrated in Figure 18, was tailored to the type of field being analyzed:

- For **string-type fields** (e.g., job title, industry category, education level), a **hybrid evaluation approach** was adopted. First, semantic similarity was computed using **Sentence Transformers** embeddings. Each similarity score was mapped to a 1–5 scale. If the similarity score fell below a defined threshold (score < 4), the result was reviewed and reclassified using an **LLM-based fallback**, allowing for contextual correction of borderline cases.
- For **list-type fields** (e.g., required skills, qualifications), a fully **LLM-based evaluation** was applied. Each model's output was compared against the reference using three qualitative metrics: **Completeness**, **Accuracy**, and **Hallucination Rate**, each rated on a 1–5 Likert scale. The evaluation was performed using structured JSON comparisons to ensure consistency and traceability.

All models were tested using two temperature settings (0.1 and 0.2) to account for generation variability, and results were segmented by country to capture linguistic and regional differences.

Across all evaluated domains, **Phi-4** emerged as the top-performing model. It consistently achieved the highest **precision, recall, and F1-scores** across job title extraction, industry classification, education level identification, and the extraction of skills and qualifications. The model demonstrated high semantic understanding and maintained stable performance across both temperature settings, while also remaining efficient in token usage.

LLaMA-3-8B also performed well, particularly in the classification of **industry** and **education levels**, offering a strong balance between semantic accuracy and computational efficiency. **Qwen-30B** showed competitive recall in several fields but lagged slightly behind in precision and had higher hallucination rates when handling list-type fields like skills.

DeepSeek-R1, in contrast, underperformed across all evaluation categories. It consistently returned lower similarity scores and exhibited weaker semantic accuracy, making it the least reliable model for metadata extraction in this context.

Given its superior overall performance and alignment with prior evaluations (e.g., in translation and deduplication), **Phi-4** was selected as the default model for the metadata extraction stage **in** the MicroIDEA pipeline. Detailed results by country, field type, and evaluation metric are presented in Appendix F, Tables 1–5.

H. Interactive Visualization

The visualization and reporting layer of the MicroIDEA platform serves as the primary interface between structured, semantically enriched job data and end users—namely policy analysts, career counselors, researchers, and job seekers. Built atop the structured outputs of the AI pipeline, this module translates complex labor market intelligence into intuitive, interactive dashboards.

Using **waiter/waitress occupations (ISCO Code 5131)** as a representative use case, MicroIDEA visualizations provide country-specific insights across Greece, Spain, and Cyprus. The dashboards are designed not only to present data but to narrate it—blending statistical graphics with AI-generated insights that support evidence-based decision-making (Figure 19).

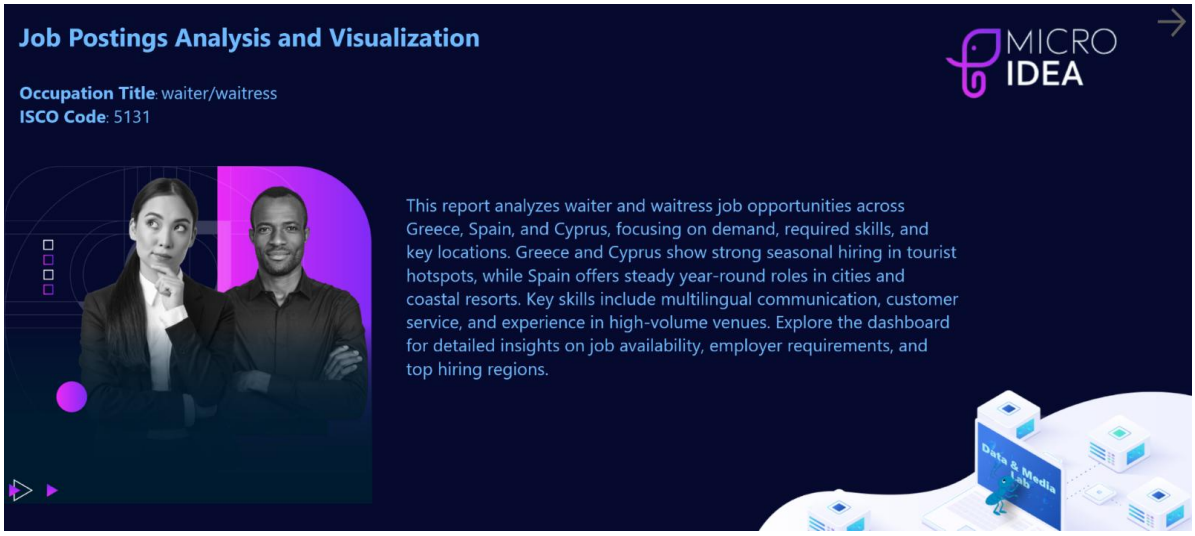

Figure 19: Intro Slide of Microidea Report

H.1 Data Sources and Integration

The dashboards are populated using structured metadata fields derived from the earlier AI-driven pipeline phases (E.1 - E.2). These include translated, deduplicated, and semantically normalized job postings. The enriched fields cover a wide range of entities—skills, qualifications, industries, languages, occupations—and are mapped to internationally recognized taxonomies such as ESCO, ISCED, NACE, and ISCO-08. All this information is continuously stored in the central JPA cloud database.

H.2 Visualization Framework and Technology Stack

The visual layer is implemented using **Microsoft Power BI**, selected for its capacity to create dynamic, cross-filtered dashboards and its compatibility with secure cloud deployment.

Power BI dashboards are deployed alongside the Django-based backend infrastructure and securely linked to real-time data via direct MariaDB connections. In the following link https://portal.micro-idea.eu/ (Menu -> Navigate -> Dashboard) showcase dynamic, auto-refreshing dashboards with integrated insights.

To enhance interpretability, each dashboard is accompanied by **text narrative panels** generated by local LLMs. These insights are created using structured prompts tailored to the visual content and embedded as dynamic commentary elements.

H.3 Dashboard Modules and Functionalities

The visualization suite is modular and organized around key analytical perspectives. Major components include:

- **Skills and Qualifications**: Bar charts and treemaps highlight dominant skills, required languages and education requirements (Figure 20,21).
- **Regional and Sectoral Trends**: Maps and pie charts show hiring activity by region (Figure 22) and by industry sector (Figure 23).

Each dashboard combines quantitative graphics with AI-generated textual insights to guide interpretation and support data-driven actions.

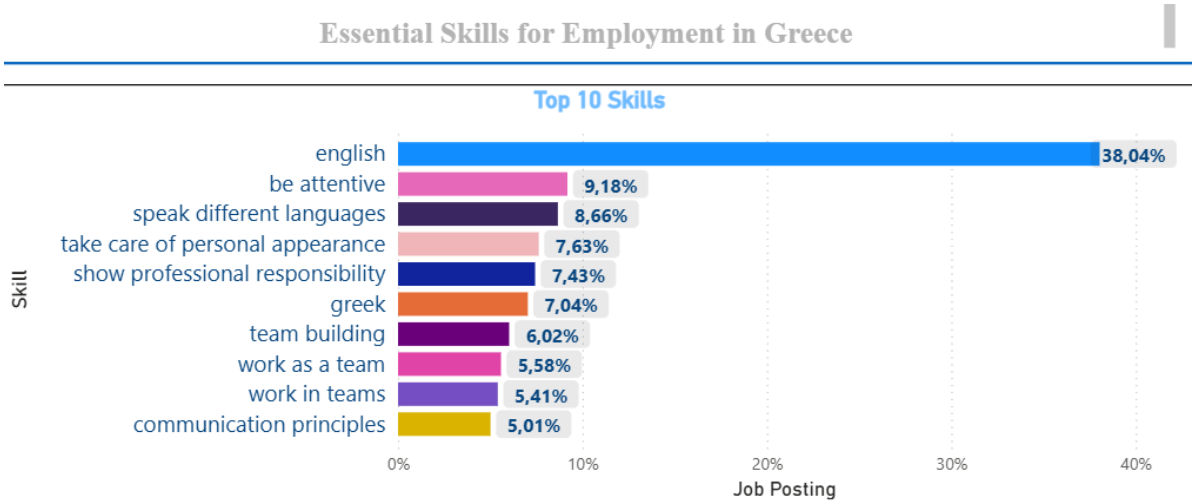

Figure 20: Top 10 Dominant Skills

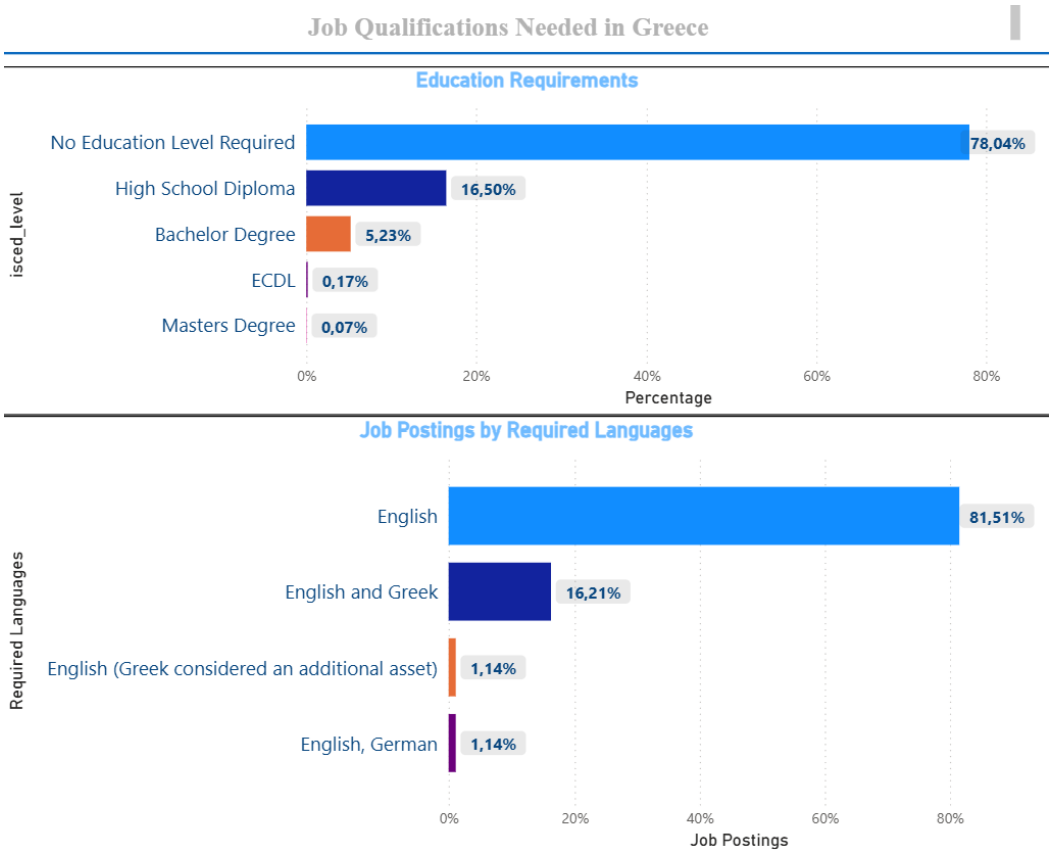

Figure 21: Required Education Level and Languages

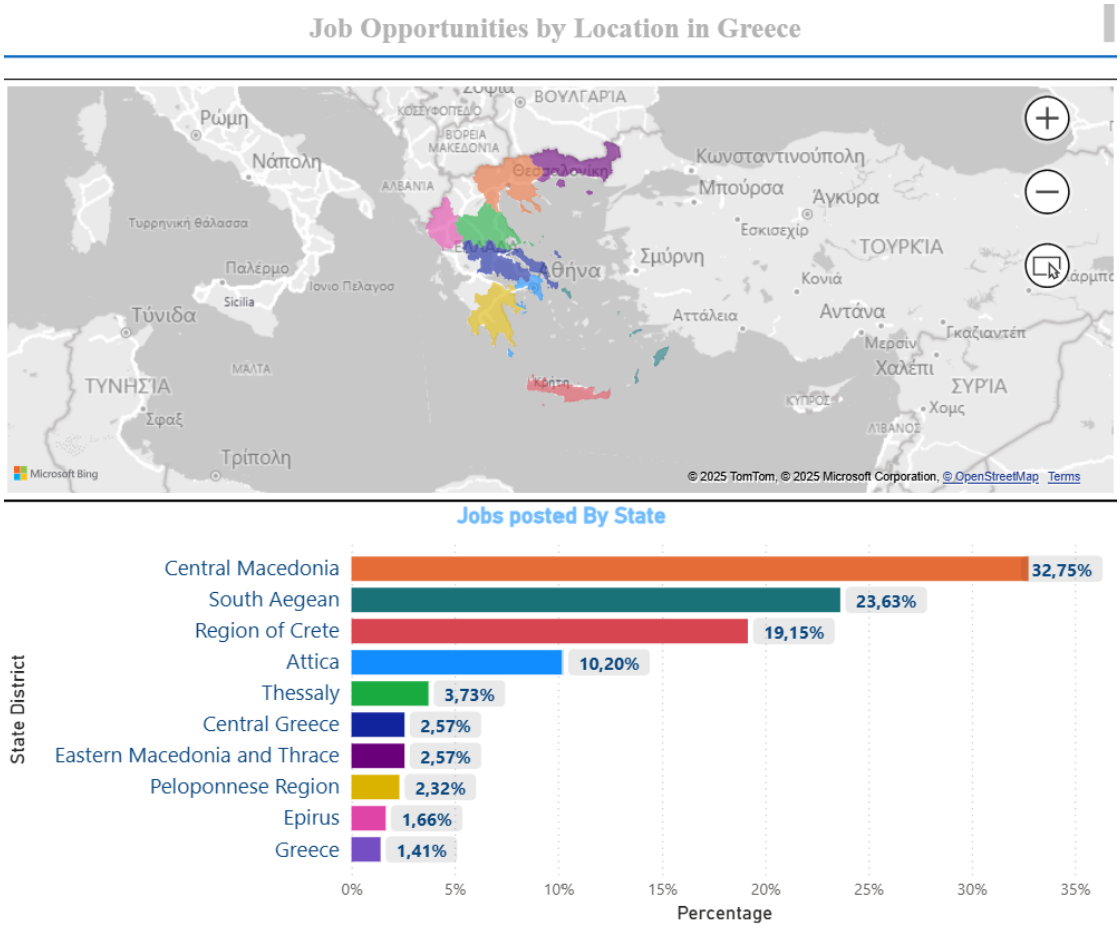

Figure 22: Job Opportunities by Location

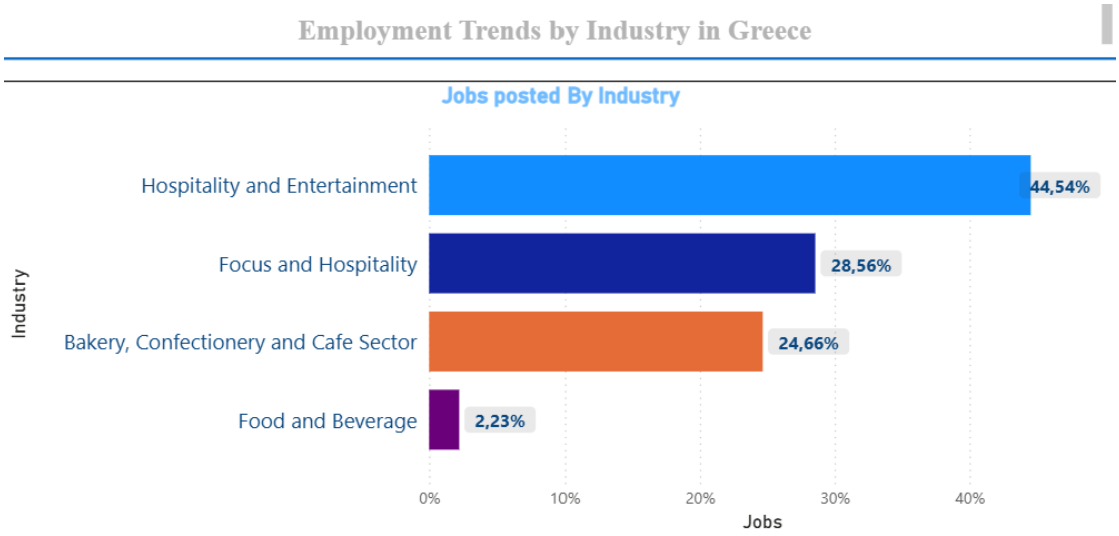

Figure 23: Job Opportunities by Industry

H.4 User Experience and Navigation Features

User navigation begins at an **index page** (Figure 24) where country and thematic filters can be selected. Navigation is supported via clickable modules and directional arrows, offering a fluid user experience.

Narrative insights are positioned next to visualizations and automatically adapt to the selected view. Icons, color schemes, and layout follow the visual identity of MicroIDEA and remain

consistent across all components. All dashboards support multilingual access and are screen-reader compatible, promoting inclusion and accessibility.

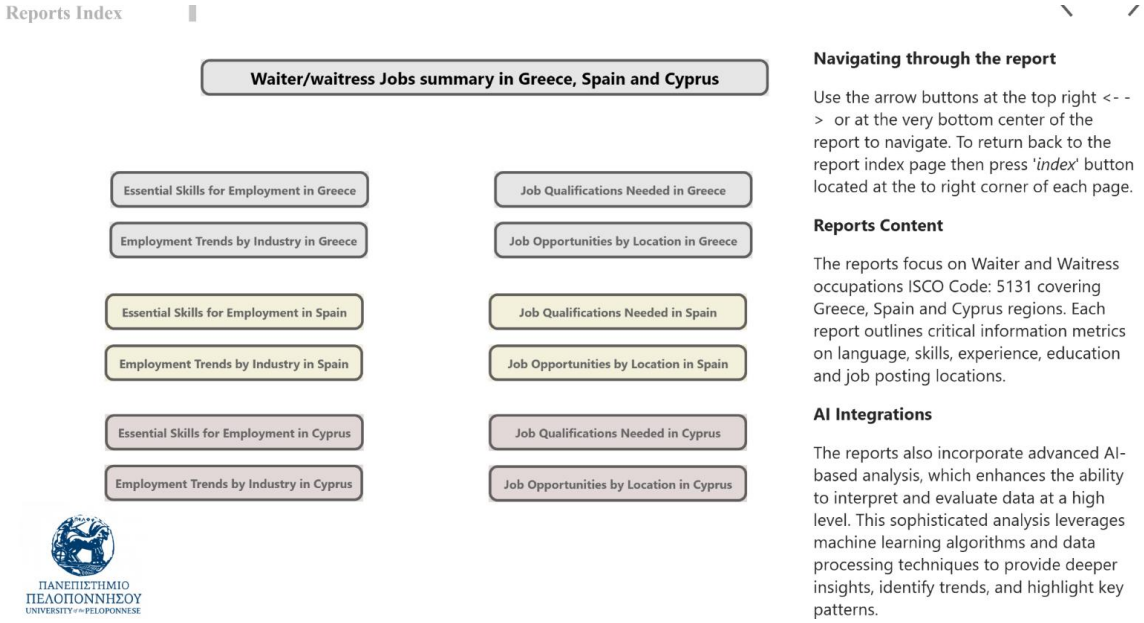

Figure 24: Reports Index Page

H.5 Interoperability and Semantic Consistency

All metadata fields visualized are semantically harmonized using the normalization phase described in E.2.2. This alignment ensures that comparisons can be made across different languages and national contexts. For example, skills in Greek, Spanish, and English are normalized to a single ESCO skill label, allowing fair cross-country comparisons.

Thanks to this semantic alignment, MicroIDEA dashboards can be easily integrated with EU labor market observatories, education/training analytics systems, and skill recommendation platforms. The LLM-generated insights use the same taxonomy-aligned inputs, ensuring full consistency across human-readable and machine-readable layers.

H.6 System Architecture and Database Design

At the backend, the entire platform is implemented using the Django framework and is composed of four tightly integrated subsystems. These are illustrated in Figure 25.

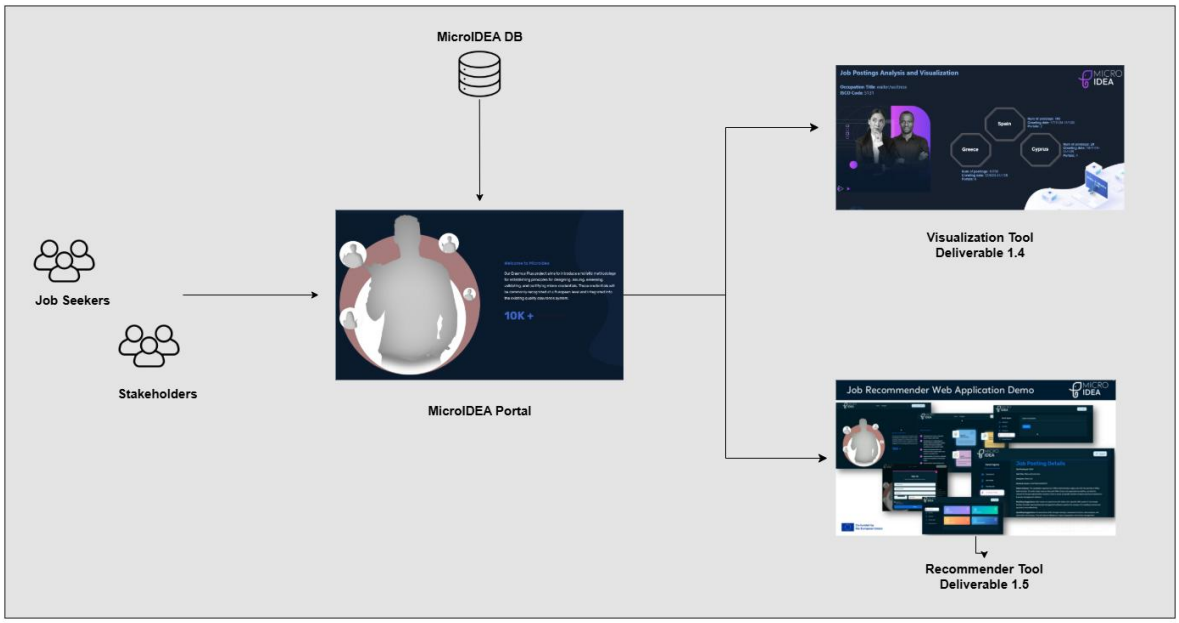

Figure 25: Visualization System Architecture

The **System Administration Interface** is designed for internal users and enables the configuration of prompt parameters, user management, and monitoring of token usage. This is the control center for maintaining operational integrity and prompt versioning.

The **Public Web Portal** Interface is a minimal front-facing component offering high-level insights, registration, and authentication functionality. Once authenticated, users are redirected to a more advanced environment.

The **User/Job Seeker Portal** provides access to a personalized dashboard where users can create and update their CVs, browse job recommendations, and track their alignment with current market needs. Recommendations are powered by semantic AI models that analyze user CVs against employer-defined requirements (experience, education, skills). Personalized prompts offer dynamic suggestions for upskilling and reskilling.

The **Jobs Portal Integration Module** continuously feeds the system with new job postings, which are processed via the AI pipeline for deduplication, translation, and entity extraction.

Supporting this architecture is a three-tiered database system:

Database	Function
Main System Database	Stores user profiles, system settings, and dashboard metadata
Chroma Vector DB	Powers semantic search and skill matching
MicroIDEA Database	Holds cleaned and enriched job postings

Table 17: Three-tiered Database System Integration

I. Conclusion and Recommendations

The MicroIDEA platform represents a major technological and methodological innovation in the field of labor market intelligence. It successfully combines cutting-edge developments in artificial intelligence with practical needs in labor market analysis, policy-making, and education. The system's architecture, rooted in an AI-powered pipeline, transforms noisy, multilingual, and unstructured job postings into harmonized, taxonomy-aligned metadata. Through processes such as cleaning, translation, semantic deduplication, LLM-based metadata extraction, and RAG-assisted normalization, the platform delivers high-quality, structured insights that can be readily used by stakeholders across sectors.

At the heart of this system is the application of Transformer-based LLMs, which have evolved dramatically in recent years—from BERT and GPT-2 to more advanced models such as GPT-4 and well performed open source LLMs, such as LLaMA-3, and Phi-4. These models are not only capable of language understanding but also of contextual reasoning, few-shot generalization, and multilingual adaptation. Their architecture, built on self-attention mechanisms, allows them to learn deep semantic relationships across diverse inputs, making them ideally suited to labor market data that vary by language, structure, and domain.

The core contribution lies in the creation of a scalable, end-to-end pipeline capable of transforming heterogeneous job data into harmonized metadata aligned with ESCO, ISCO, NACE, and ISCED standards. Through robust preprocessing, semantic deduplication, and multilingual translation, the platform ensures linguistic neutrality and removes redundancy. AI-driven metadata extraction and normalization ensure precision, contextual understanding, and semantic alignment, while the integrated visualization layer delivers actionable insights for stakeholders across education, employment, and policy domains.

Evaluation results across all components—translation, deduplication, and metadata extraction—confirm the effectiveness of the selected AI models and methodologies, especially the Phi-4 model, which demonstrated the highest consistency across tasks and languages. The iterative evaluation strategy, grounded in human-validated test sets and LLM-as-a-Judge methodology, provides a replicable and auditable quality assurance framework.

To ensure the long-term relevance and impact of the MicroIDEA platform, several strategic steps are recommended. First, regular evaluation and updating of AI models—particularly those used for translation, metadata extraction, and normalization—should be institutionalized to keep pace with evolving labor market language and emerging occupational terms. This process should leverage updated ground truth datasets and explore fine-tuning on domain-specific content to maintain high semantic fidelity. Second, expanding the system's geographic coverage beyond Greece, Cyprus, and Spain would enhance the analytical depth of the platform and enable more comprehensive cross-country comparisons, supporting EU-wide policy objectives and mobility strategies. Third, collaboration with maintainers of classification taxonomies like ESCO, ISCO, and NACE is essential to address gaps identified during normalization. Many newly observed skills and job titles—especially in digital and green sectors—are not yet represented in these taxonomies. MicroIDEA can contribute valuable insights and curated examples to help extend these standards, ensuring they remain current and reflective of real labor market dynamics across Europe.

J. Appendices

Appendix A – Raw Input Data per Country

Field Name	Type	Cyprus	Spain	Greece	Description
id	int(11)	Yes	Yes	Yes	Unique identifier for each job posting
foreign_key_id	int(11)	Yes	Yes	Yes	The id in the portal table
source	varchar(255)	Yes	Yes	Yes	Name of the portal
title	varchar(255)	Yes	Yes	Yes	Title of the job posting
location	varchar(255)	Yes	Yes	Yes	Geographic location of the job
company	varchar(255)	Yes	Yes	Yes	Name of the company posting the job
posted	varchar(255)	Yes	Yes	Yes	Date the job was posted
scraped	datetime	Yes	Yes	Yes	Date the data was scraped from the source
description	text	Yes	Yes	Yes	Detailed description of the job
type	varchar(255)	Yes	-	Yes	Type of employment (e.g., full-time, part-time)
url	varchar(255)	Yes	Yes	Yes	URL of the job posting

deadline	varchar(255)	Yes	-	-	Application deadline for the job
salary	varchar(255)	-	Yes	-	Salary for the job
experience	varchar(255)	-	Yes	-	Required experience level
category	varchar(255)	Yes	-	-	Job category or industry
level	varchar(255)	-	Yes	-	Level of the job position
sector	varchar(255)	-	Yes	-	Economic sector of the job
requirements	text	-	Yes	-	Specific job requirements
contract	varchar(255)	-	Yes	-	Type of contract offered

Appendix B

Table B1: Metadata Field Definition version 1

Field Name	Type	Description
Advertising company	str	The name of the company or agency responsible for managing the recruitment process. This may be different from the seeking company, particularly if an external recruitment agency is used.
hiring company	str	The name of the company that is actively seeking to fill the position. This is the organization where the successful candidate will be employed.
Location city or area	str	Specific city or area of the job location
job title	str	The official title of the position as described in the job posting. This should clearly represent the specific role and level within the organization, including any relevant specializations or focus areas.

		Examples include 'Senior Data Scientist,' 'Marketing Manager,' or 'Junior Software Developer specializing in Frontend Development.
occupation	str	The broader category or profession to which the job belongs. This should generalize the role, focusing on the type of work performed rather than specific titles. For example, for 'Senior Machine Learning Engineer,' the occupation could be 'Software Engineer' or 'Machine Learning Specialist.' If the occupation cannot be determined, default to using the job title as a fallback option..
job season	str	If applicable, the time period or season for the job.
job responsibilities	List[str]	Tasks and responsibilities associated with the job.
required skills	List[str]	Abilities, proficiencies, or competencies required to perform a task, including language skills, soft skills (like communication) or technical skills (like programming).
qualifications	List[str]	previous experience, working experience,degrees, certifications, licenses, or formal requirements (like a degree or driving license).
communication contact	str	The email address or phone number to send applications to.
salary details	str	Details about the salary or compensation offered.
Required education level	str	State the minimum education requirement (e.g., High School Diploma, Bachelor’s Degree, Master’s Degree)
employment type	str	The type of employment (e.g., Full-time, Part-time, Contract).
job benefits	List[str]	Additional benefits provided by the employer. This may include health insurance, retirement plans, performance bonuses, paid time off, and other non-monetary perks that enhance the overall compensation package.
job positions	int	The total number of job vacancies available for the position. If the exact number is not stated, infer the number of positions based on context and typical hiring practices in the industry, but default to 1 if uncertain.

working hours	str	Expected working hours or shifts.
Working experience required	str	If exists in the description, The amount of time spent working in a specific field, industry, or occupation that is relevant to the job being applied for
industry category	str	The industry or sector the job belongs to.
required languages	List[str]	Language requirements for the job.
Remote work option	str	Whether the job is Remote, Hybrid, or On-site.
Hiring company description	str	Brief information about the hiring company or organization, including its main services or products, target audience, company mission, and any notable qualities that make it unique or appealing as an employer.
Work environment details	str	Information about the working conditions or environment.
travel requirements	str	If travel is required, specify the frequency or extent.
provided equipment	str	Any tools, equipment or resources provided by the company.

Table B2: Metadata Field Definition version 2

Field Name	Type	Description
Advertising company	str	The name of the company or agency responsible for managing the recruitment process. This may be different from the seeking company, particularly if an external recruitment agency is used.
hiring company	str	The name of the company that is actively seeking to fill the position. This is the organization where the successful candidate will be employed.

Location city or area	str	Specific city or area of the job location.
job title	str	The official title of the position as described in the job posting. This should clearly represent the specific role and level within the organization. Examples include 'Senior Data Scientist,' 'Marketing Manager,' or 'Junior Software Developer specializing in Frontend Development.'
occupation	str	The broader category or profession to which the job belongs. This should generalize the role, focusing on the type of work performed rather than specific titles. For example, for 'Senior Machine Learning Engineer,' the occupation could be 'Software Engineer' or 'Machine Learning Specialist.' If the occupation cannot be determined, default to using the job title as a fallback option..
job season	str	If applicable, the time period or season for the job.
job responsibilities	List[str]	Key duties and tasks that the candidate will be expected to perform as part of the role. These describe what the employee will do on a day-to-day or regular basis.
required skills	List[str]	Specific abilities, technical proficiencies, or knowledge areas that a candidate must possess to perform the job effectively. This includes Hard skills (e.g., Python, Microsoft Excel, AutoCAD), Soft skills (e.g., teamwork, communication, adaptability), Language skills (e.g., 'Fluent in English', 'Conversational French') unless framed as a formal/legal requirement
qualifications	List[str]	Educational credentials, certifications, degrees, or other formal requirements that a candidate must meet to be eligible for the job (e.g., 'Bachelor's degree in Computer Science', 'CPA certification').
communication contact	str	The email address or phone number to send applications to.
salary details	str	Details about the salary or compensation offered.
Required education level	str	The minimum formal education required for the job, if specified. This may include levels such as 'High School Diploma', 'Associate's Degree', 'Bachelor's Degree', 'Master's Degree', or 'No formal education required'. If the description refers to a specific type of school (e.g., 'School of Tourism'), extract the equivalent education level if possible.

employment type	str	The type of employment (e.g., Full-time, Part-time, Contract).
job benefits	List[str]	Perks or advantages offered by the employer in addition to salary. Examples include health insurance, paid time off, retirement plans, remote work options, bonuses, or wellness programs.
job positions	int	The total number of job vacancies available for the position. If the exact number is not stated, infer the number of positions based on context and typical hiring practices in the industry, but default to 1 if uncertain.
working hours	str	Expected working hours or shifts.
Working experience required	str	The minimum amount or type of prior professional experience required for the role. This may be expressed in years or as prior experience in a specific domain or job function (e.g., '3+ years of experience in software development').
industry category	str	The broader industry or sector in which the job exists. This could include labels like 'Information Technology', 'Healthcare', 'Finance', 'Education', or 'Retail'.
required languages	List[str]	Any specific languages the candidate must speak, write, or understand to perform the job (e.g., 'English', 'Spanish', 'Mandarin').
Remote work option	str	Whether the job is Remote, Hybrid, or On-site.
Hiring company description	str	Brief information about the hiring company or organization, including its main services or products, target audience, company mission, and any notable qualities that make it unique or appealing as an employer.
Work environment details	str	Information about the working conditions or environment.
travel requirements	str	If travel is required, specify the frequency or extent.

provided equipment	str	Any tools, equipment or resources provided by the company.

Appendix C

Table C1: ESCO Skills/Competencies lexicon - Sample

conceptType	conceptUri	skillType	reuseLevel	preferredLabel	altLabels	description
KnowledgeSkillCompetence	http://data.europa.eu/esco/skill/0005c151-5b5a-4a66-8aac-60e734beb1ab	skill/competence	sector-specific	manage musical staff	manage staff of music;coordinate duties of musical staff;manage music staff;direct musical staff;;;Staff Coordination;Task Management;Music Arranging;Music Preparation;Vocal Coaching Supervision;Team Assignment;Music Scoring Management;Music Copying Coordination	Assign and manage staff tasks in areas such as scoring, arranging, copying music and vocal coaching.
KnowledgeSkillCompetence	http://data.europa.eu/esco/skill/00064735-8fad-454b-90c7-ed858cc993f2	skill/competence	occupation-specific	supervise correctional procedures	oversee prison procedures;manage correctional procedures;monitor correctional procedures;manage prison procedures;monitor prison procedures;oversee correctional procedures;	Supervise the operations of a correctional facility or other correctional procedures, ensuring that they are compliant with legal regulations, and ensure that the staff complies with regulations, and aim to improve the facility's

						efficiency and safety.

Table C2: ESCO/ISCO Occupations Lexicon - Sample

ESCO Code	Occupation Title	ISCO Code	Alternative Occupation Titles	Occupation Description
1101	air force officer	110	royal airforce officer, flight lieutenant, flight officer, air commodore, air force officer, military group captain, command and control officer, pilot officer, squadron leader	Air force officers specialise in flying or ground duties, and supervise a team of air force personnel. They coordinate their team's training and welfare, and perform duties specific to their area of specialisation.
1102	armed forces officer	110	major, lieutenant general, Roal Air Force officer, second lieutenant, general, lieutenant, military officer, air force officer, Royal Navy officer, major general, RAF officer, armed forces officer, captain, officer cadet, colonel, naval officer, army officer, Royal Marines officer, navy officer, lieutenant colonel, brigadier	Armed forces officers supervise operations and manoeuvres, assign duties, and command subordinate staff. They ensure efficient communication within and between units and perform training duties. They also operate equipment and supervise equipment maintenance.

Table C3: NACE Industry Lexicon - Sample

Class	Title	Description
1.11	Growing of cereals, other than rice, leguminous crops and oil seeds	This class includes all forms of growing of cereals, leguminous crops and oil seeds. Includes the growing of these plants for the purpose of seed production. The growing of these crops is often combined within agricultural units. This class includes: - growing of cereals, e.g.: - wheat - grain maize - sorghum - barley - rye - oats - millets - pseudocereals, fruits or seeds - used as cereals, e.g.: - quinoa - amaranth - chia - growing of leguminous crops, e.g.: - beans - broad beans - chick peas - cow peas - lentils - lupines - peas - pigeon peas -

		growing of oil seeds, e.g.: - soya beans - groundnuts - castor bean - linseed - mustard seed - niger seed - rapeseed - safflower seed - sesame seed - sunflower seed This class excludes: - growing of rice, see 01.12 - growing of sweet corn, see 01.13 - growing of maize, lupines, kale for fodder, see 01.19 - growing of oleaginous fruits, see 01.26
1.12	Growing of rice	This class includes: - growing of rice
1.13	Growing of vegetables and melons, roots and tubers	This class includes: - growing of leafy or stem vegetables, e.g.: - artichokes - asparagus - cabbages - cauliflower and broccoli - lettuce and chicory - spinach - growing of fruit bearing vegetables, e.g.: - cucumbers and gherkins - eggplants (aubergines) - tomatoes - watermelons - growing of root, bulb or tuberous vegetables, e.g.: - carrots - turnips - garlic - onions (incl. shallots) - leeks and other alliaceous vegetables - growing of mushrooms and truffles - growing of vegetable seeds, including sugar beet seeds, excluding other beet seeds - growing of sugar beet - growing of chillies, peppers (Capsicum spp.) - growing of roots and tubers, e.g.: - potatoes - sweet potatoes - cassava - yams - swedes and mangolds This class also includes: - growing of sweet corn This class excludes: - spices and aromatic crops, see 01.28 - growing of mushroom spawn, see 01.30

Appendix D – Translation Evaluation Results

Table D1 - Quality Eval Metrics - Greek to English

Metric	Argos	DeepSeek-R1	Mistral-7b	Phi-4
Semantic Accuracy	3.51	4.12	3.63	4.31
Fluency	2.8	4.11	3.81	4.11
Overall Score	3.16	4.12	3.72	4.21

Table D2 - Quality Eval Metrics - Spanish to English

Metric	Argos	DeepSeek-R1	Mistral-7b	Phi-4
Semantic Accuracy	3.65	4.36	4.61	4.60
Fluency	2.71	4.16	4.29	4.31
Overall Score	3.18	4.26	4.45	4.46

Table D3 - Performance Metrics - Greek to English

Method	Time(seconds)/Request	Tokens/Request
Argos	1.19	N/A
DeepSeek-R1	3.33	1025
Mistral-7b	3.65	1027
Phi-4	3,34	1027
GPT-4	3,13	645

Table D4 - Performance Metrics - Spanish to English

Method	Time(seconds)/Request	Tokens/Request
Argos	1.48	N/A
DeepSeek-R1	3.45	661
Mistral-7b	3.35	660
Phi-4	3,32	661
GPT-4	3,59	621

Appendix E – Deduplication Evaluation Results

Model	Precision	Recall	Accuracy	F1-Score	Time(s)	Tokens
Phi-4	0.95	0.96	0.94	0.96	129	999
DeepSeek-R1	0.95	0.96	0.94	0.96	143	997
Llama-3-8b	0.91	0.91	0.88	0.91	75	1029
Mistral-7b	0.82	1.00	0.85	0.90	131	1170

Appendix F – Metadata Extraction Evaluation Results

Table F1: Evaluation of Job Title Table : Job Title - Greece

Model	Temp	Similarity	Semantic	Precision	Recall	F1-Score	Tokens/Req
phi-4	0.1	0.85	4.30	0.90	0.97	0.94	97.16
phi-4	0.2	0.84	4.28	0.92	0.95	0.94	95.01
qwen-30b	0.1	0.86	4.34	0.87	0.99	0.93	82.71
qwen-30b	0.2	0.86	4.35	0.88	0.99	0.93	78.45

DeepSeek-R1	0.1	0.63	3.53	0.90	0.71	0.79	164.27
DeepSeek-R1	0.2	0.62	3.46	0.88	0.70	0.78	170.71
llama-3-8b	0.1	0.82	4.24	0.93	0.90	0.91	93.41
llama-3-8b	0.2	0.81	4.22	0.92	0.89	0.90	93.22

Table F2: Evaluation of Industry Category

Model	Temp	Similarity	Semantic	Precision	Recall	F1-Score	Tokens/Req
phi-4	0.1	0.81	4.30	0.93	0.99	0.96	104.60
phi-4	0.2	0.82	4.33	0.93	1.00	0.96	105.33
qwen-30b	0.1	0.75	4.02	0.84	0.98	0.90	133.76
qwen-30b	0.2	0.76	4.04	0.84	0.97	0.90	125.43
DeepSeek-R1	0.1	0.68	3.79	0.87	0.87	0.87	151.81
DeepSeek-R1	0.2	0.69	3.79	0.87	0.90	0.88	152.08
llama-3-8b	0.1	0.72	3.93	0.88	0.92	0.90	141.15
llama-3-8b	0.2	0.74	4.01	0.90	0.94	0.92	127.90

Table F3: Evaluation of Education Level

Model	Temp	Similarity	Semantic	Precision	Recall	F1-Score	Tokens/Req
phi-4	0.1	4.06	3.87	1.43	3.96	746.56	phi-4
phi-4	0.2	4.04	3.83	1.52	3.91	754.33	phi-4
qwen-30b	0.1	3.75	3.61	1.82	3.64	755.21	qwen-30b
qwen-30b	0.2	3.84	3.58	1.78	3.67	760.20	qwen-30b

DeepSeek-R1	0.1	2.95	2.84	1.60	2.86	747.00	DeepSeek-R1
DeepSeek-R1	0.2	2.97	2.86	1.51	2.89	741.50	DeepSeek-R1
llama-3-8b	0.1	3.58	3.43	1.54	3.46	757.73	llama-3-8b
llama-3-8b	0.2	3.57	3.44	1.56	3.47	753.68	llama-3-8b

Table F4: Evaluation of Required Skills

Model	Temp	Completeness	Accuracy	Hallucinations	Overall	Tokens/Req
phi-4	0.1	4.06	3.87	1.43	3.96	746.56
phi-4	0.2	4.04	3.83	1.52	3.91	754.33
qwen-30b	0.1	3.75	3.61	1.82	3.64	755.21
qwen-30b	0.2	3.84	3.58	1.78	3.67	760.20
DeepSeek-R1	0.1	2.95	2.84	1.60	2.86	747.00
DeepSeek-R1	0.2	2.97	2.86	1.51	2.89	741.50
llama-3-8b	0.1	3.58	3.43	1.54	3.46	757.73
llama-3-8b	0.2	3.57	3.44	1.56	3.47	753.68

Table F5: Evaluation of Qualifications

Model	Temp	Completeness	Accuracy	Hallucinations	Overall	Tokens/Req
phi-4	0.1	4.13	3.95	1.23	4.06	659.55
phi-4	0.2	4.24	4.06	1.37	4.12	644.07
qwen-30b	0.1	3.98	3.78	1.63	3.83	658.43
qwen-30b	0.2	4.03	3.79	1.66	3.85	657.14
DeepSeek-R1	0.1	3.07	3.01	1.41	3.01	656.86
DeepSeek-R1	0.2	3.09	3.05	1.40	3.06	655.11
llama-3-8b	0.1	4.23	4.01	1.88	4.04	670.13
llama-3-8b	0.2	4.20	3.88	1.98	3.93	666.82

List of Tables

List of Figures

References

1. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., et al (2020). Language Models are Few-Shot Learners. Retrieved from arxiv: http://arxiv.org/abs/2005.14165.
2. Yang, J., Wang, M., Zhou, H., Zhao, C., Yu, Y., Zhang, W., & Li, L. (2019). Towards Making the Most of BERT in Neural Machine Translation. ArXiv.
3. Skondras, P., Zotos, N., Lagios, D., Zervas, P., Giotopoulos, K., & Tzimas, G. (2023). Deep Learning Approaches for Big Data-Driven Metadata Extraction in Online Job Postings. Information.
4. Baldwin, T., Clarke, W., Garcia de Macedo, M. M., de Paula, R. A., & Das, S. (2022). Better Skill-based Job Representations, Assessed via Job Transition Data. 2022 IEEE International Conference on Big Data (Big Data), 2182-2185.
5. Reimers, N., Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Retrieved from arxiv: http://arxiv.org/abs/1908.10084.
6. Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Retrieved from arxiv: http://arxiv.org/abs/1810.04805.
7. Khokhlova, O., Khokhlova, A. N., & Choyzhalsanova, A. (2022). Development of an Algorithm to Analyze Vacancies in the Labor Market Based on Open-Source Data. Voprosy statistiki.
8. Zheng, X., Zhang, C., & Woodland, P. (2021). Adapting GPT, GPT-2 and BERT Language Models for Speech Recognition. 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), 162-168.
9. Raiaan, M., Mukta, Md., Fatema, K., Fahad, N. Sakib, S., et. al. (2024). A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges. *IEEE Access*, 26839-26874.
10. Zhao, Y., Chen, H., & Mason, C.M. (2021). A Framework for Duplicate Detection from Online Job Postings. IEEE/WIC/ACM International Conference on Web Intelligence, 249-256.
11. Engelbach, M., Klau, D., Klintz, M., & Ulrich, A. (2024). Combining Embeddings and Domain Knowledge for Job Posting Duplicate Detection. Retrieved from arxiv: http://arxiv.org/abs/2406.06257.
12. Li, N., Kang, B., & De bie, T. 2025. LLM4Jobs: Unsupervised occupation extraction and standardization leveraging Large Language Models. Knowledge-Based Systems, 113302.
13. Choung, H.W., Hou, L., Longpre, S., Zoph, B., Tay, Y., et al (2022). Scaling Instruction-Finetuned Language Models. Retrieved from arxiv: http://arxiv.org/abs/2210.11416.
14. Alsayed, N., & Awad, W. (2023). A framework for Labor Market Analysis using Machine Learning. 2023 International Conference on IT Innovation and Knowledge Discovery (ITIKD), 1-5.
15. Malandri, L., Mercorio, F., & Serino, A. (2025). SkiLLMo: Normalized ESCO Skill Extraction through Transformer Models. Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing.
16. Zheng, L., Chiang, W., Sheng, Y., Zhuang, S., Wu, Z., et al (2023). Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. Retrieved from arxiv: http://arxiv.org/abs/2306.05685.
17. Finlay, P. (2020). Argos Translate. Retrieved from Argos Translate: https://github.com/argosopentech/argos-translate.
18. Abdin, M., Aneja, J., Behl, H., Bubeck, S., Eldan, R, et al (2024). Phi-4 Technical Report. Retrieved from arxiv: http://arxiv.org/abs/2412.08905.
19. Miller, D.L. (2024). WordLlama: Recycled Token Embeddings from Large Language Models. Retrieved from https://github.com/dleemiller/wordllama.
20. Lavi, D., Medentsiy, V., Graus, D. (2021). conSultantBERT: Fine-tuned Siamese Sentence-BERT for Matching Jobs and Job Seekers. Retrieved from arxiv: http://arxiv.org/abs/2109.06501.
21. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M., et al (2023). LLaMA: Open and Efficient Foundation Language Models. Retrieved from arxiv: http://arxiv.org/abs/2302.13971.

22. Green, T., Maynard, D., Lin, C., Calzolari, N., Bechet, F., et al (2022). Development of a Benchmark Corpus to Support Entity Recognition in Job Descriptions. Proceedings of the Thirteenth Language Resources and Evaluation Conference.
23. Yan, J., Yan, P., Chen, Y., Li, J., Zhu, X., Zhang, Y. (2024). Benchmarking GPT-4 against Human Translators: A Comprehensive Evaluation Across Languages, Domains, and Expertise Levels. Retrieved from arxiv: http://arxiv.org/abs/2411.13775.
24. Makwana, D., Engineer, P., Dabhi, A., Chudasama, H. (2023). Sampling Methods in Research: A Review. International Journal of Trend in Scientific Research and Development (IJTSRD).
25. Skondras, P., Zervas, P., Tzimas, G. (2023). Generating Synthetic Resume Data with Large Language Models for Enhanced Job Description Classification. Future Internet, 363.
26. Powers, D.M.W. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. Retrieved from arxiv: http://arxiv.org/abs/2010.16061
27. DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., et al (2025). DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. Retrieved from arxiv: http://arxiv.org/abs/2501.12948.
28. Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S. (2023). Mistral 7B. Retrieved from arxiv: http://arxiv.org/abs/2310.06825.
29. Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., et al (2025). Qwen3 Technical Report. Retrieved from arxiv: http://arxiv.org/abs/2505.09388

MICRO IDEA

MICRO-credentials
Identifying,
DEveloping, testing and
Assessing innovative approaches

Project Partners

Scan for more

micro-idea.eu

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.