

Article

Ontology-Based Semantic Normalization of Resumes for Classification

Victor-Valentin Anghel ¹, Theodor Borangiu ^{1,2} , Silviu Răileanu ^{1,*} and Cătălin Negulescu ¹

¹ Department of Automation and Industrial Informatics, National University of Science and Technology POLITEHNICA Bucharest, 060042 Bucharest, Romania; victor.anghel@stud.acs.upb.ro (V.-V.A.); theodor.borangiu@upb.ro (T.B.); catalin.negulescu@stud.electro.upb.ro (C.N.)

² Academy of Romanian Scientists, 050045 Bucharest, Romania

* Correspondence: silviu.raileanu@upb.ro

Abstract

During the recruitment process, it is possible for CVs to appear well-organized. However, it is not always straightforward to compare them. The same competence may be denoted by different designations, and the levels of competence are not universally employed in the same manner. Natural Language Processing (NLP) methodologies can extract these data points; however, ensuring the consistency of this data across multiple CVs remains a challenge. In a multitude of cases, the comparability of two profiles remains ambiguous. In the present study, an ontological approach is adopted to solve this issue. The concept under discussion is that of the extraction of entities from CVs and their subsequent representation in a more structured form, utilizing RDF and an ontology aligned with ESCO—the multilingual classification of European Skills, Competences, and Occupations. Subsequently, the rules of SHACL are applied to verify the semantic coherence of the data; the validated data are transmitted to a model for classification. At this stage, the dataset becomes smaller, but semantically cleaner, more traceable, and enriched with validation indicators that can be used by the classification model. The proposed system is implemented as a set of microservices. A Spring Boot component coordinates the flow, whilst the Python services, implemented using Python 3.10.12 are responsible for the primary processing stages including extraction, validation and classification. A same-corpus ablation was conducted to separate ontology-guided profile selection from the contribution of the validation-derived quality features. On the same 35,770 filtered CV–job pairs, adding these features increased external benchmark accuracy from 0.794 to 0.809, recall from 0.760 to 0.865, F1-score from 0.749 to 0.786, and ROC-AUC from 0.881 to 0.887. A *p*-value of 0.00540 paired with a 1.54 effect ratio from McNemar’s test showed a statistically significant paired difference between the two configurations. However, precision decreased from 0.739 to 0.720 while Average Precision compressed from 0.851 down to 0.844. Rather than scaling performance uniformly across the entire evaluation suite, the ontology layer acts as a targeted traceability and semantic refinement filter that contributes information beyond filtered-profile selection alone and produces a metric-dependent change in classifier behaviour at the validation-selected threshold.



Academic Editor: Wu Deng

Received: 14 May 2026

Revised: 8 July 2026

Accepted: 16 July 2026

Published: 20 July 2026

Copyright: © 2026 by the authors.

Published by MDPI on behalf of the International Institute of Knowledge Innovation and Invention. Licensee MDPI, Basel, Switzerland. This article

is an open access article distributed under the terms and conditions of the

[Creative Commons Attribution](#)

(CC BY) license.

Keywords: ontology-based systems; semantic validation; SHACL; knowledge graphs; CV classification; ESCO; Natural Language Processing; microservices architecture

1. Introduction

In recruitment, resumes usually come in formats like Europass or LinkedIn. These formats are structured and, at first glance, they seem easy to process automatically. Skills are listed clearly, and sometimes candidates also provide a level for each of them. However, once we try to compare different resumes, things become less consistent.

In practice, the same skill can appear under slightly different names, depending on how the candidate chooses to describe it. The meaning of proficiency levels is also not always clear. What one person considers advanced might not be the same for someone else. On top of that, relationships between skills are rarely defined explicitly. Because of these differences, comparing candidates in an automatic way becomes harder than expected.

The difficulties encountered in CV processing extend beyond simple terminology differences. In practice, the same extracted information may present a variety of semantic issues. A technology can be assigned to an inappropriate category, a proficiency level may be absent or expressed in an unclear form, and some extracted statements may not be supported by a traceable fragment of text in the original document.

Many validation pipelines handle these situations through a strict pass-or-fail decision. Profiles that satisfy all validation requirements continue through the workflow, while those that fail are excluded from further processing. Although this strategy helps remove clearly problematic data, it may also eliminate profiles that remain largely informative despite containing a limited number of recoverable issues.

This distinction is particularly important in recruitment data. Candidate profiles are often incomplete, inconsistently formatted, or only partially detailed. A missing level declaration or an uncertain concept mapping does not necessarily make the entire profile unusable. In many cases, the remaining information still provides meaningful evidence about the candidate's background and qualifications.

For this reason, the proposed approach introduces the notion of training eligibility. Rather than treating all validation failures identically, the ontology layer separates profiles into different categories according to the severity of the detected issues. Fully conforming profiles follow the standard processing path, profiles containing critical semantic deficiencies are excluded, and profiles with recoverable inconsistencies remain available for subsequent stages.

When a profile belongs to this recoverable category, the associated uncertainty is preserved explicitly through ontology-derived indicators instead of being hidden or ignored. These indicators are then passed to the classifier and to the downstream decision-support components, allowing later stages of the recruitment workflow to take the quality of the extracted information into account when producing recommendations and matching decisions.

There has been a lot of progress in recent years in using NLP and machine learning for extracting information from resumes. Recent studies have explored the potential of transformer-based architecture and large language models for recruitment-related information extraction and semantic matching [1]. These studies have highlighted the scalability of these methods and the challenges related to explainability and semantic consistency. The methods work well when it comes to identifying patterns in texts. Still, they mostly operate at a statistical level and leave some gaps. For example, similar skills are not always aligned in the same way, and checking whether the extracted data is consistent is not that easy. In many situations, it is also unclear why a certain decision was made.

For this reason, some recent approaches have started to combine these techniques with more structured representations. Ontologies and knowledge graphs are often used for this purpose. Using an ontology changes the way the data is handled. Instead of treating skills as simple text labels, they can be described more clearly and connected to each

other [2–4]. Previous work has used ontologies, taxonomies, and knowledge graphs for skills representation, semantic matching, labour-market recommendation, and candidate–job comparison. However, in many cases, the ontology remains mainly a representation or matching layer. Less attention has been given to how ontology validation results can be operationalized as learning signals for a downstream classifier. This is helpful when we try to keep the data consistent, not just extract it. It also makes it easier to apply validation rules and spot cases where something does not fit.

In this work, we follow this direction and introduce an ontology-based approach for recruitment. The system combines automatic extraction of information from resumes and job descriptions with domain ontology aligned to the ESCO vocabulary. The ontology is validated using SHACL (Shapes Constraint Language) rules. In this setup, ESCO is used mainly as reference vocabulary, while the ontology is used to represent skill statements, normalized levels, and job requirements in a more structured form. These elements remain connected to the original text, so the source can still be checked if needed. After this step, the data is stored in a knowledge graph from where we derive a set of features that are easier to interpret compared to standard representations. These features are then used by a machine learning classification model implemented in the CV classification service. The goal is not only to improve the candidate–job matching, but also to make the process easier to understand, especially from the point of view of decision transparency [5].

2. Related Work

The recruitment domain has been examined in the literature through several complementary research directions. Ontology-based approaches support the semantic representation of CVs and job descriptions, while ESCO-oriented NLP methods focus on skill extraction and taxonomy alignment. Labour-market knowledge graphs further model relations among jobs, skills, occupations, and learning resources. In parallel, SHACL and ShEx-based studies provide mechanisms for checking data conformance and generating constraints whereas neural and hybrid person–job fit models mainly address pairwise matching and ranking. More recent explainable, recruiter-in-the-loop, and agentic systems extend this line of work toward workflow support, candidate screening, and adaptive assessment generation.

2.1. Ontology-Based CV/Job Matching Systems

Ontology-based recruitment systems have long been used to represent CVs, job descriptions, skills, and candidate requirements in a more explicit and interoperable way [6,7]. The approach summarized in [6] addresses CV screening and CV search for a given IT job description by extracting and comparing skill graphs from both the CV and the job specification. It proposes dedicated ontologies for representing CV and job content and combines these semantic representations with NLP and machine learning techniques to reduce ambiguity in text. Similarly, the ICT-domain system in [7] combines NLP, machine learning, fuzzy logic, and an ontology-based skill representation, with explicit modelling of proficiency levels and an overall matching score between candidate profiles and job requirements. These works show that semantic representations and proficiency-level modelling are useful for reducing ambiguity in recruitment matching.

Beyond ontology-based CV/job representation, knowledge graphs extend semantic modelling toward broader recommendation scenarios. JobEdKG [8] connects jobs, skills, and online courses in a knowledge graph designed for course recommendation and for predicting skills required by target roles. A relevant aspect of JobEdKG is its explicit modelling of uncertainty, where facts in the graph receive scores based on their prevalence in the data sources. Skills2Job [9] recommends occupations from a user-provided set

of skills by combining graph-based storage, Revealed Comparative Advantage, word embedding, and ESCO-based similarity. CareerBERT [10] also uses ESCO and EURES resources but focuses on matching resumes to general ESCO job titles in a shared embedding space for generic job recommendations. These systems demonstrate that graph structures, taxonomy alignment, and embedding spaces can support recommendation beyond simple keyword search.

The present approach builds on ontology-based and graph-based recruitment systems, but it assigns a more operational role to ontology. The ontology is not used only as a representation layer, a matching vocabulary, or a general labour-market graph. It is placed between the NER/JSON extraction stage and the XGBoost classifier as a semantic-control layer. Extracted CV/job fields are converted into RDF claims, aligned where possible to local concepts or ESCO concepts, marked through verification indicators, validated through SHACL, and then transformed into ontology-derived signals. These signals include conformance status, recoverable non-conformance, critical violation count, trainable violation count, unknown violation count, unverified concept count, and unverified concept ratio. Therefore, the classifier can receive not only class-level similarity features, but also information about the semantic reliability of the extracted evidence.

2.2. NLP and Transformer-Based Methods for Resume Parsing and Skill Normalization

NLP-based resume parsing is a necessary step in recruitment systems because CVs and job descriptions are usually available as noisy, semi-structured, or unstructured text. In this context, ESCO-oriented NLP work is particularly relevant because it focuses on extracting skill mentions and normalizing them against controlled labour-market taxonomies. Clavie and Soulie [11] study large language models as zero-shot ESCO skill matchers and show the importance of candidate generation and reranking when mapping job-posting text to ESCO skills. This is important because skill mentioned in recruitment documents may be ambiguous, incomplete, or expressed through variants that do not directly match a taxonomy label.

ESCOXLM-R [12] follows a different but related direction by using multilingual taxonomy-driven pre-training over ESCO. Its objective is to improve job-market NLP tasks such as skill extraction, skill classification, and occupation-related classification. This confirms that labour-market language benefits from domain-specific and taxonomy-aware representation learning, especially in multilingual settings where the same skill or occupation may appear in different forms across documents.

These studies are relevant to the extraction and normalization stage of the present system. The proposed method also starts with an unstructured CV and job text, but the extracted information is organized into six semantic classes: programming languages, foreign languages, certifications, work experience, soft skills, and education. The role of the NLP component is to produce structured JSON fields and evidence spans, which are then converted into RDF claims. However, the system does not assume that extraction alone is sufficient. Because NER and transformer-based extraction may produce uncertain, incomplete, or incorrectly normalized entities, the extracted claims are further checked through the ontology and SHACL validation layer before being used by the supervised classifier.

2.3. SHACL Validation and Constraint Generation in Knowledge Graphs

The use of SHACL is grounded in the broader literature on RDF graph validation. The study summarized in [13] examines SHACL and ShEx as constraint languages for validating knowledge graphs, combining empirical evidence from academic and industrial settings with an analysis of available tools and automatic rule-derivation methods. Its conclusions are relevant for this work because they highlight the limits of fully manual

validation-rule design and the need for systematic or semi-automatic validation approaches in heterogeneous data scenarios.

Astrea [14] is also important because it automatically generates SHACL validation rules from ontology constraint patterns, covering both value constraints and structural constraints. This supports the idea that validation rules can be derived or maintained from ontological models rather than written entirely by hand. Pareti et al. [15] complement this foundation with a formal analysis of SHACL language features, recursive semantics, and the satisfiability and containment decision problems for different SHACL fragments. The W3C SHACL Recommendation [16] defines SHACL as a language for validating RDF graphs against conditions expressed as shapes, and the SHACL review by Pareti and Konstantinidis [17] discusses the language from data validation to schema reasoning, including practical implementations and formal aspects. Together, these works justify SHACL as a principled mechanism for checking whether RDF data conforms to a set of semantic and structural constraints.

The contribution of the present system is not simply the use of SHACL for RDF validation. The important difference is how the SHACL report is operationalized inside the recruitment-learning pipeline. Validation is executed before the supervised classifier, and the report is parsed into task-specific categories: critical violations, trainable violations, and unknown violations. These categories are then combined with unverified concept statistics to identify non-conforming profiles that can still be treated as recoverable. Training eligibility is derived from this distinction: a profile is retained for training if it either fully conforms to the SHACL shapes or belongs to the recoverable non-conforming subset. As a result, SHACL validation becomes a bridge between semantic data quality and downstream learning, rather than a standalone validation step that only accepts or rejects RDF data.

2.4. Neuro-Symbolic and Hybrid Symbolic–Data-Driven Matching Approaches

In parallel with ontology-based and graph-based approaches, person–job fit research has developed neural and hybrid models that learn semantic representations from resumes, job postings, and historical recruitment interactions. APJFNN [18] models person–job fit through an ability-aware neural architecture that uses attention mechanisms to represent job requirements and candidate experiences. PJFCANN [19] extends this direction by combining local semantic representations based on co-attention with global experience representations derived from recruitment history through graph neural networks.

ConFit v2 [20] is particularly relevant among recent dense retrieval approaches. It addresses sparse interaction labels in resume–job datasets by using hypothetical reference resumes generated by a large language model and runner-up hard-negative mining. It evaluates resume–job matching on real-world datasets and reports improvements over prior baselines in ranking tasks. This line of work demonstrates the strength of modern representation learning for resume–job retrieval and ranking. Professional-network-based person–job fit models further show that social and contextual information can enrich matching beyond the resume–job text pair [21].

The present work does not introduce a new dense neural encoder and does not compete directly with neural retrieval models. Instead, it follows a hybrid symbolic–data-driven strategy. The NLP component extracts structured semantic fields from CVs and job descriptions. The ontology and SHACL layer then normalize and validate these fields, producing explicit semantic quality indicators. These ontology-derived indicators are combined with class-level similarity features and passed to an XGBoost classifier. In such cases, the score must be read together with the validation output. A weak association may come from an actual mismatch between the candidate profile and the job requirements, but it may also reflect elements that were only partially confirmed during extraction or

normalization. For this reason, the ontology layer keeps track of these signals and passes them as evidence-quality indicators for the supervised classifier.

2.5. Explainable Recommendation and Recruiter-in-the-Loop Systems

Explainability is important in recruitment because a final compatibility score is not enough for recruiter decision-making. Work on LLM-based screening also follows this direction, using contextual information from the candidate profile and the job description to produce structured assessments that can be inspected by recruiters [22]. This indicates that, in recruitment, explanation is not only a technical addition to the model, but part of the decision-support process itself. General-purpose explanation methods such as SHAP [23] can show how individual features contribute to a model prediction, and recent recruitment systems increasingly combine semantic retrieval with explainable AI [24].

Industrial platforms provide additional context for recruiter-in-the-loop and agentic recruitment workflows. Phenom presents X+ Ontologies as an intelligence layer for matching, skill inference, agentic workflows, and AI-generated guides [25]. Its 2026 hiring automation benchmark points to a similar practical issue: screening, assessment, scheduling, and credential verification are still not fully connected in many recruitment workflows [26]. This supports the need for more integrated systems. However, the available industrial descriptions do not provide a reproducible method in which SHACL validation reports are transformed into classifier features and traceable semantic explanations.

The proposed approach treats explainability at two levels. First, the classifier result is interpreted by semantic class, such as programming languages, foreign languages, certifications, work experience, soft skills, and education. Second, the explanation can be brought closer to the actual CV and job content by looking at concrete elements inside these classes. For example, the system can show whether Java, Python, Docker, English B2, an AWS certification, or a required number of experience years supported or weakened the matching score. This type of explanation is easier to use than a single global score because it points to the exact requirements that influenced the decision. The same element-level representation could also support future adaptive assessment mechanisms, but this component is not implemented or evaluated in the present study.

To make the comparative positioning clearer, Table 1 summarizes representative related works according to the main dimensions discussed in this section. The comparison shows that previous studies usually address only part of the problem: ontology-based representation, ESCO-oriented normalization, graph-based recommendation, SHACL validation, neural matching, or explainable ranking. By contrast, the proposed approach combines ontology-based semantic validation, EvidenceSpan-based traceability, recoverable uncertainty handling, supervised classification, and element-level explanation within the same microservice pipeline.

As shown in Table 1, the proposed approach does not claim novelty from using ESCO, SHACL, knowledge graphs, or supervised classification as isolated components. Its contribution lies in the way these elements are connected. The ontology layer is placed between extraction and classification, where it validates RDF claims, records evidence quality, separates critical and recoverable inconsistencies, and converts these validation outcomes into features used by the XGBoost classifier. This makes semantic validation operational rather than merely descriptive and allows the classification result to be interpreted together with the reliability and textual provenance of the extracted evidence.

Table 1. Comparative positioning of the proposed approach against related work.

Work	Focus	ESCO Support	SHACL Validation	EvidenceSpan Traceability	Uncertainty Handling	Semantic Validation
Ontology-based CV/job matching [6]	Semantic CV/job representation and skill-graph matching	No	No	No	Partial	Yes
ICT ontology and fuzzy matching system [7]	Ontology-based skill representation with proficiency levels	No	No	No	Partial	Yes
JobEdKG [8]	Knowledge graph for jobs, skills, and course recommendation	No	No	No	Yes	Partial
Skills2Job [9]	Graph-based occupation recommendation from user skills	Yes	No	No	Partial	Partial
SHACL and ShEx validation study [13]	RDF graph validation and constraint languages	No	Yes	No	No	Yes
Astrea [14]	Automatic generation of SHACL shapes from ontology patterns	No	Yes	No	No	Yes
SHACL formal and practical studies [15–17]	SHACL semantics, constructs, validation, and schema reasoning	No	Yes	No	No	Yes
ESCO skill matching with LLMs [11]	ESCO-oriented skill normalization using LLMs	Yes	No	No	Partial	Partial
ESCOXLM-R [12]	Multilingual taxonomy-driven pre-training for job-market NLP	Yes	No	No	Partial	Partial
CareerBERT [10]	Resume-to-ESCO occupation matching in embedding space	Yes	No	No	No	Partial
APJFNN [18]	Neural person–job fit modelling with attention	No	No	No	Partial	No
PJFCANN [19]	Co-attention and graph neural network model for person–job fit	No	No	No	Partial	No
ConFit v2 [20]	Dense retrieval for resume–job matching with hard negatives	No	No	No	Partial	No
Professional-network-based person–job fit [21]	Matching using professional and contextual network information	No	No	No	Partial	No
LLM-based screening and explainable recruitment systems [22,24]	Explainable screening and recruiter decision support	Partial	No	Partial	Partial	Partial
Industrial recruiter-in-the-loop platforms [25,26]	Agentic recruitment workflows and skill inference	Partial	Not specified	Not specified	Partial	Partial
Proposed approach	Ontology-guided validation, semantic quality features, XGBoost matching, and traceable semantic explanations.	Yes	Yes	Yes	Yes	Yes

The novelty of the proposed approach consists in operationalizing ontology-based validation within a microservice-based recruitment pipeline: semantic constraints are not used only to accept or reject RDF data, but to produce evidence-quality signals for XGBoost classification and granular explanations of CV–job matching. The experimentally evaluated contribution concerns the use of conformance status, training eligibility, recoverable non-conformance, violation counts, unverified concept ratio, and semantic confidence as additional classifier features. Recruiter-guided score adjustment and adaptive assessment are not part of the evaluated contribution and are retained only as possible future application-level extensions.

The resulting architecture is modular, loosely coupled, traceable, and scalable across the extraction, validation, matching, and explanation services.

3. System Architecture and Processing Pipeline

The proposed system uses an ontology as a central step for semantic post-processing of the information extracted from resumes. The main goal here is to validate and normalize the JSON data structure that is later sent to the classification service. In this way, the machine learning model receives data whose semantic validation status, recoverable inconsistencies, and associated uncertainty are made explicit before classification. This helps limit the propagation of errors, especially those coming from the document structuring microservice, whose role is to extract the semantic categories defined.

The ontology layer isolates and addresses four distinct semantic anomalies:

- Cross-class errors occur when an extraction algorithm misallocates an entity, such as classifying Java as a foreign language rather than a programming technology;
- Proficiency-level inconsistencies emerge when the identified concept lacks a valid, explicit, or aligned vocabulary marker;
- Evidence inconsistencies represent claims completely detached from a verifiable EvidenceSpan within the source document;
- Taxonomy-alignment failures pinpoint local concepts unable to achieve a confident mapping path to ESCO or local ontologies.

Mitigation depends strictly on severity; the system applies automated corrections, flags borderline instances with explicit uncertainty markers, or enforces total exclusion from the final training set.

The processing flow starts with basic text pre-processing applied to CV documents. This includes removing less relevant lexical elements, such as conjunctions and prepositions. After that, a named entity recognition (NER) model is applied. The model picks up six types of information that are usually relevant in the recruitment process of IT specialists: programming languages, foreign languages, certifications, work experience, soft skills, and education. It also keeps some details about where these appear in the original text, using start and end positions.

Once this part is done, the entities go to the ontology layer. At this point, they are organized in a more structured way and linked through semantic relations. The data is then checked using SHACL rules. This step is used to make sure the structure is correct and that each element can still be traced back to the original CV. In practice, each item is connected to an EvidenceSpan attribute, which points to its position in the text. A profile is considered fully conforming only when the relevant concept mappings, proficiency levels, EvidenceSpan references, and structural properties satisfy the validation rules. Recoverable non-conforming profiles are retained with explicit uncertainty indicators.

After this validation step, the enriched semantic representation is passed to the CV classification service. The XGBoost model uses the six matching dimensions together with the ontology-derived quality indicators to estimate the compatibility of each CV–job pair.

The ontology validation stage remains active during processing; its constraints are not learned implicitly by XGBoost and are not replaced by the classifier during inference.

The full processing flow is shown in the diagram presented in Figure 1, where the architecture includes a structuring service that extracts relevant entities from resumes using a fine-tuned BERT model.

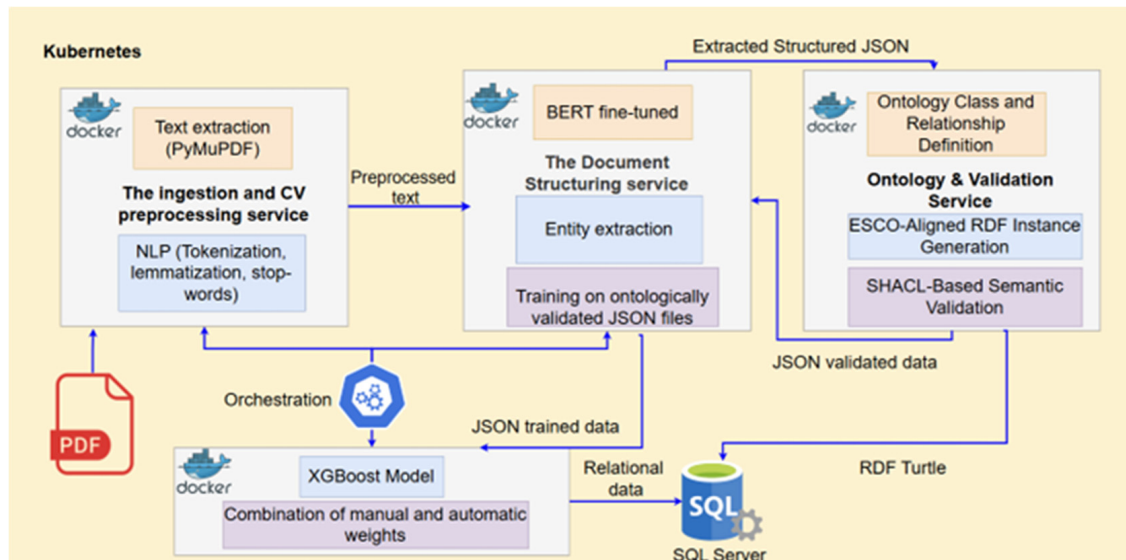


Figure 1. Processing workflow.

Along with the extracted features, the model also returns the positions where the information appears in the text, together with an associated confidence score.

The extracted information is then sent to a dedicated microservice for ontology handling and semantic validation. This service is responsible for mapping the identified entities into a conceptual form. In practice, it instantiates the semantic representation of the six categories of interest using the classes defined in the OWL ontology.

For each extracted entity, corresponding ontology instances are created such as SkillClaim (statements about declared skills), SkillConcept (canonical domain concepts), ProficiencyLevel (normalized skill levels), and EvidenceSpan (which points to the exact position of the information in the source document).

This representation is then checked using SHACL rules, so that the data remains consistent and each statement can still be traced back to the original CV. The relationships among the extracted entity, the contextual claim, the canonical concept, the proficiency level, and the supporting EvidenceSpan are summarized in Figure 2.

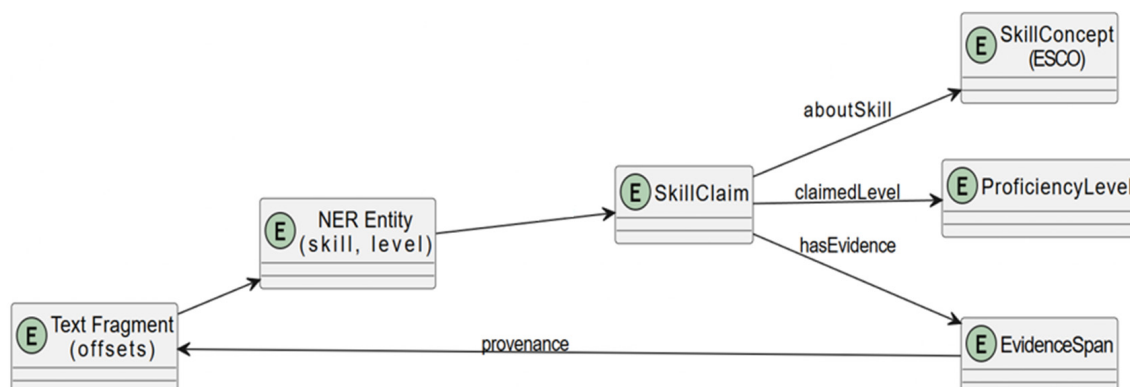


Figure 2. OWL modelling.

The architecture separates entity extraction, semantic validation, and classification into independent services. This modularity allows the ontology, SHACL rules, and external vocabularies to evolve without requiring corresponding changes in the other processing components. At the same time, normalized concepts and EvidenceSpan links preserve traceability from the classifier inputs to the source document. The ontology service can therefore be reused in other workflows that require the same semantic-control function.

4. Ontology and Validation Service

The ontology and validation service acts as a link between the operational data model of the application and the semantic representation used during validation and CV classification.

For clarity, the validation states used by the ontology service are defined as follows. A profile is considered valid when all generated RDF claims conform to the SHACL constraints, have valid EvidenceSpan references, and use accepted concept and level values. A profile is invalid when at least one mandatory semantic requirement is violated. The architecture isolates invalid profiles by partitioning them into recoverable and non-recoverable categories. A profile qualifies as recoverable if its specific anomalies allow for normalization, remapping, or uncertainty tagging without compromising the core semantic integrity. Operationally, this mitigation applies to cross-class assignments requiring structural reindexing, missing taxonomic levels that accept a standardized default value, or localized concepts where rich textual evidence compensates for a lack of direct ESCO alignment. A non-recoverable profile contains errors that make the extracted representation unreliable for training, such as missing EvidenceSpan values, invalid offsets, absent mandatory concepts, or semantic mappings below the accepted similarity threshold.

At the concept level, a verified concept is a concept that matches either the local ontology vocabulary or an ESCO concept with sufficient confidence. An unverified concept is a concept that remains supported by textual evidence but cannot be confidently aligned with ESCO or the local vocabulary. Unverified concepts are not silently discarded. They are retained with an explicit *isUnverified* marker and contribute to the unverified concept ratio used as a semantic confidence signal.

Figure 3 shows the data structure used in the system. It is built around entities like Candidates, Candidate_CVS, Jobs, Job_Applications, and CV_Classification_Results. Among these, the Candidate_CVS entity has a central role. It stores both the raw text of the CV and the JSON structure produced by the document structuring service. This JSON is later used as input for the ontology service.

The semantic fields inside json_CV, for example programming languages, foreign languages, certifications, education, work experience, and soft skills, are extracted and then mapped to the corresponding classes in the OWL ontology. For each detected element RDF instances are created, such as SkillClaim, EducationClaim, CertificationClaim, or WorkExperienceClaim. These instances link the candidate to more general domain concepts.

The relations between the Candidates, Candidate_CVS, and Job_Applications tables make it possible to connect these ontology instances to a specific candidate and, later, to a particular job application context. Because of this, the ontology service does not run separately. It stays connected to the application data model and helps keep a clear link between the stored data and its semantic representation.

Based on these RDF instances, the service applies SHACL validation rules. The goal here is to check several aspects at once. This includes the structure of the data, the presence of textual evidence through EvidenceSpan, the validity of skill levels (ProficiencyLevel), and whether the mapping to canonical concepts is correct, including alignment with ESCO.

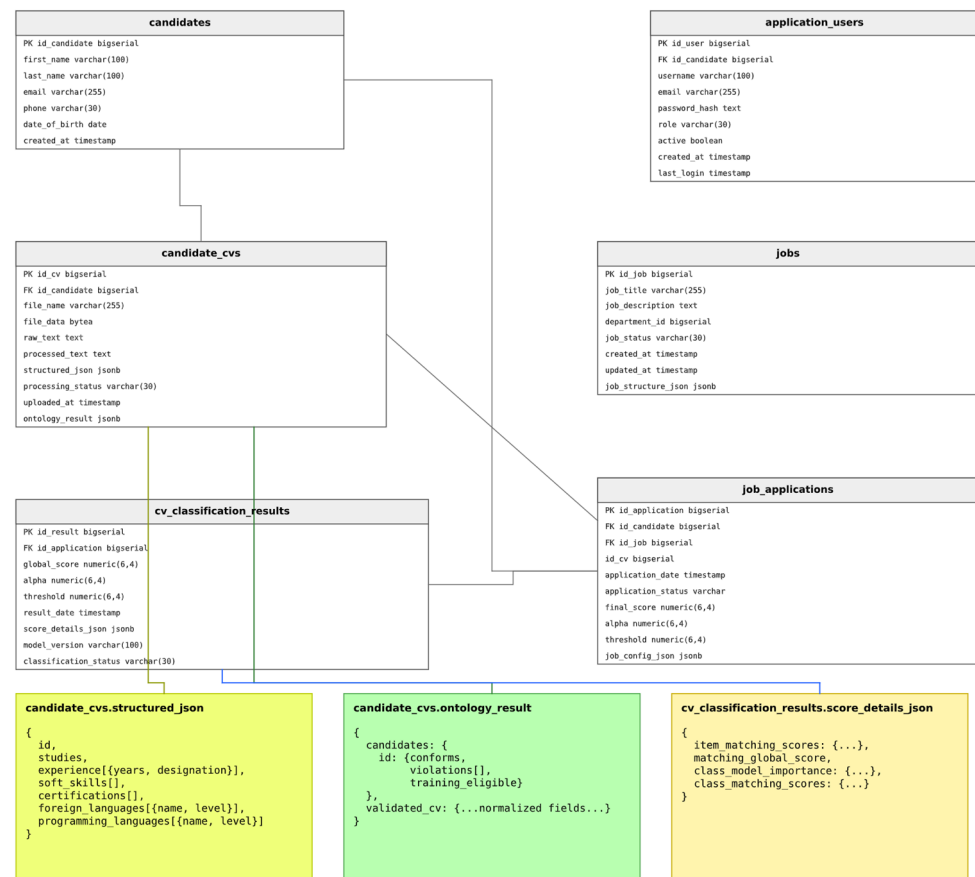


Figure 3. Database schema and JSON structures used by the CV-job matching pipeline.

During this step, some inconsistencies can appear. For example, a programming language might end up classified as a foreign language. Instead of rejecting the entire CV, the ontology service handles these cases more gradually and applies a tolerant validation strategy. It adjusts the semantic representation of the extracted elements. Some elements do not fit well in the initial category. In those cases, they are either moved to a more suitable one, if that makes sense or the corresponding concepts are retained but explicitly marked as unverified. This design choice preserves potentially useful information while exposing its uncertainty.

Importantly, the ontology does not enforce a strict accept/reject decision. Instead, it produces enriched semantic signals, including validation status, inconsistency indicators, and a training eligibility flag that reflects whether the identified deficiencies are recoverable. As a result, non-conforming but training-eligible profiles are not discarded but are instead propagated to the classification stage.

The decision regarding training eligibility is derived from two sources of information: the SHACL validation results and the outcome of the concept verification process. The ontology service begins by examining whether the RDF representation satisfies all validation constraints defined in the ontology. Profiles that satisfy all constraints are considered semantically valid and can be used directly during training.

When one or more constraints are violated, the ontology service does not immediately reject the profile. Instead, the detected issues are separated according to their impact on semantic reliability. The first group contains critical violations. These include situations where textual evidence is missing, EvidenceSpan offsets cannot be validated, mandatory concepts are absent, or concept mappings fall below the minimum similarity threshold accepted by the system. Such cases are considered non-recoverable because the semantic interpretation of the profile cannot be trusted.

A second group contains violations that do not necessarily invalidate the profile. Examples include missing proficiency levels, ambiguous level descriptions, cross-class assignments that can be corrected through ontology mapping, and concepts that cannot be fully verified but are still supported by evidence extracted from the original document. These situations introduce uncertainty, but the underlying information remains potentially useful.

Violations that cannot be clearly assigned to either category are recorded separately and later examined as part of the error-analysis process. This allows new error patterns to be identified without changing the validation workflow.

A profile is considered recoverable non-conforming when no critical violations are present and all remaining issues belong to the trainable category. Such profiles are not treated as fully compliant, yet they are not removed from the training process either. Instead, the uncertainty associated with them is preserved and made available to the classification model through ontology-derived indicators.

For this purpose, the ontology service generates a set of semantic quality features describing the validation outcome. These include the overall conformance status, the training eligibility flag, the recoverable non-conforming indicator, the number of detected violations, the distribution of violations by category, the number of unverified concepts, the proportion of unresolved concepts, and an aggregated semantic confidence score. In this way, the classifier receives not only information about candidate qualifications, but also information about the reliability of the extracted semantic representation itself.

These additional signals are leveraged by the classification model as indicators of semantic confidence. In particular, the proportion of unverified concepts is used to derive a penalty factor, allowing the model to down-weight unreliable information without ignoring it entirely. At the same time, the identification of missing competencies enables the system to capture partial matches between candidates and job requirements.

The functional integration of training eligibility ensures that the model encounters datasets aligned with the structural noise of actual human resources. Whereas a strict filter eliminates non-compliant profiles over minor and remediable issues producing a pristine but unrepresentative training set, the proposed methodology intentionally captures these boundary configurations. Incorporating valid profiles that carry explicit uncertainties including normalized levels or unverified, evidence-supported competencies accustoms the classifier to partial, noisy, and remediable cases. Rather than relying on idealized inputs, the system's adaption to these boundary conditions mirrors actual recruitment pipelines. In these environments, an ambiguous extracted skill seldom prompts a systemic rejection, provided the surrounding profile yields sufficient contextual evidence.

The ontology-derived quality indicators are passed directly to the classifier as additional input features. Their contribution is evaluated through the same-corpus ablation reported in Section 6.3, where two models are trained on identical filtered CV–job pairs, with the only difference being the inclusion or exclusion of these indicators. This experimental design separates the predictive contribution of the validation report from the effect of selecting a cleaner training corpus.

The ontology is not equivalent to a dictionary or a synonym list. A dictionary can map surface forms such as “js” to “Javascript” or “postgres” to “PostgreSQL”, but it does not represent the type of the concept, the relation between a claim and its textual evidence, the expected proficiency level, or the validation constraints associated with the claim.

In the proposed ontology, an extracted entity is represented as a typed RDF claim linked to a canonical concept, an EvidenceSpan, and, when relevant, a ProficiencyLevel. This allows the system to reason over the class of the concept and detect inconsistencies. When Java is extracted as a ForeignLanguageClaim with a CEFR level B1, the ontological

layer intercepts the mismatch by verifying that the concept belongs to `ProgrammingLanguageConcept` rather than `ForeignLanguageConcept`.

This capability marks the boundary between lexical matching and semantic validation. A standard synonym dictionary merely recognizes “Java” as an isolated term, lacking the structural framework required to evaluate whether the claim type, level type, and underlying evidence relations maintain semantic coherence.

The ontology itself uses a simple three-part model. This makes it easier to separate the domain concepts from the statements extracted from documents and from the text fragments that support them. In practice, this model is built around three elements:

- **Concept (canonical concept)**—refers to a stable notion in the domain, such as “Python”, “English”, “AWS Certified Solutions Architect” or “Software Engineer”. In the ontology, these appear as instances of concept-related classes and can be aligned with external vocabularies like ESCO.
- **Claim (statement)**—captures the fact that a candidate mentions a certain concept in a CV, sometimes together with a level, a duration, or other details. This is the part that is checked during validation.
- **EvidenceSpan (text evidence)**—points to the part of the document where the information was found, using start and end offsets. It can also include a confidence score. This element is important for traceability and helps when explanations are needed.

4.1. Ontology Classes

The microservice maps the six semantic categories extracted by the NER model into ontology classes. This mapping can be seen as a simple triplet of the form (canonical_concept, SHACL_constraint, concept_claim).

- **Programming languages:**
 - canonical_concept: `ProgrammingLanguageConcept`
 - SHACL_constraint: `claimedLevel`, `confidence`, `hasLevel`
 - concept_claim: `ProgrammingLanguageClaim`
- **Foreign languages:**
 - canonical_concept: `ForeignLanguageConcept`
 - SHACL_constraint: `claimedLevel`, `cefrLevel`, `hasLevel`
 - concept_claim: `ForeignLanguageClaim`
- **Certifications:**
 - canonical_concept: `CertificationConcept`
 - SHACL_constraint: `issuedBy`, `issueDate`, `confidence`, `hasEvidence`
 - concept_claim: `CertificationClaim`
- **Education:**
 - canonical_concept: `EducationConcept`
 - SHACL_constraint: `degreeLevel`, `confidence`, `hasEvidence`
 - concept_claim: `EducationClaim`
- **Work experience:**
 - canonical_concept: `WorkRoleConcept`
 - SHACL_constraint: `role`, `duration`, `hasEvidence`
 - concept_claim: `WorkExperience`
- **Soft skills:**
 - canonical_concept: `SoftSkillConcept`
 - SHACL_constraint: `claimedLevel`, `confidence`, `hasEvidence`
 - concept_claim: `SoftSkillClaim`

At the same time, two additional classes are defined, namely ProficiencyLevel and EvidenceSpan. The first one is used to map textual values to numerical ones through normalization. For example, foreign language proficiency levels can be represented as A1—0.17, A2—0.33, B1—0.50, B2—0.67, C1—0.83, C2—1.00. The second one is used to store the entity recognition score together with its position inside the CV.

The CEFR proficiency levels, ranging from A1 to C2, are represented in the system through numerical values distributed within the interval (0,1]. The purpose of this transformation is not to model the exact educational distance between consecutive CEFR levels. Rather, it provides a simple numerical representation that preserves the natural ordering of the scale and makes it possible to compare language requirements during feature construction and candidate–job matching.

The mapping scale converts proficiency levels into ascending numerical values, anchoring the maximum weight at C2 as the ceiling of the CEFR framework. Conversely, the baseline A1 level receives a small positive increment rather than zero, preserving its mathematical significance within the vector calculations. Although it represents the lowest proficiency level, it still reflects the presence of a declared language competence and therefore should contribute to the matching process.

This representation should be viewed as an ordinal encoding designed for machine learning purposes. It allows language proficiency information to be incorporated into similarity calculations and classification features while maintaining the relative ordering of the original CEFR categories. The encoding is not intended to provide a linguistic interpretation of CEFR progression, nor does it assume that the distance between adjacent levels is identical in educational terms.

The ontology also relies on ESCO as an external reference vocabulary for validating and normalizing extracted concepts from the six semantic categories. ESCO alignment is applied after local normalization. The system first checks whether the extracted entity is present in a local domain vocabulary containing common IT technologies, programming languages, certifications, soft skills, and education terms. This local step is important for ambiguous terms that frequently occur in IT recruitment.

For example, Java is interpreted as a programming language when it appears in a technical-skill context or near terms such as Spring, backend, JVM, SQL, Docker, or developer. It is not accepted as a foreign language unless the surrounding evidence explicitly indicates a natural-language context.

If no local concept is found, the system searches for candidate ESCO concepts. The extracted label and the ESCO preferred and alternative labels are normalized through lowercasing, punctuation removal, and token normalization. A lexical similarity score is then computed between the extracted expression and the candidate ESCO labels. To secure structural accuracy, the validation framework conditions mapping acceptance on two concurrent criteria: the primary candidate must clear the established similarity threshold, and the margin separating the top two candidates must satisfy a strict differential constraint. Whenever this inter-candidate delta narrows, the system flags the configuration as ambiguous. Instead of executing a forced assignment to an erroneous ESCO URI, the architecture preserves the concept under an unverified status.

Manual verification is used for concepts that appear frequently as unverified or ambiguous. Such concepts are reviewed and either added to the local vocabulary, mapped to an ESCO concept, or kept as local domain concepts when ESCO coverage is insufficient. This prevents the system from incorrectly connecting ambiguous or emerging IT terms and acknowledges that ESCO may not cover all new technologies or role-specific terms.

4.2. Ontology Rules

Once the ontology classes are defined, a set of SHACL rules is needed to check the semantic validity, quality, and consistency of the data before it is used for training in the classification microservice [27,28].

The SHACL constraints were designed manually from the ontology model, the target JSON schema, and the observed error patterns produced by the NER and structuring services. They were not automatically generated in the current implementation. The rules cover the six semantic categories used in the system: programming languages, foreign languages, certifications, work experience, soft skills, and education. Additional shapes validate EvidenceSpan and ProficiencyLevel instances. The constraints check mandatory links, accepted value ranges, concept-class compatibility, level validity, evidence traceability, and ESCO/local concept alignment.

Beyond merely filtering out invalid RDF graphs, these validation rules fundamentally diagnose the underlying semantic inconsistencies. The resulting SHACL report undergoes structured parsing to segment anomalies into actionable streams: critical violations that block processing, trainable variations suitable for model exposure, and unknown violations requiring manual inspection. This categorization makes it possible to evaluate the impact of validation on training eligibility and classification.

The rules cover the six main categories, including certifications, foreign languages, education, soft skills, programming languages, and work experience. They are designed to handle cases such as non-standard values or situations where there is ambiguity between classes. These cases are not ignored. They are marked explicitly and handled during processing. This allows the pipeline to filter out data that does not match the expected structure, while keeping some additional information that could be useful later for improving the XGBoost model used in the classification service.

Given the rules outlined in Table 2, an example of a cross-class error is as follows: SHACL detects that the aboutConcept property does not belong to the ForeignLanguageConcept class, as it is assigned to one of the other five classes; the statement is marked as non-compliant, and a correction will be applied to assign it to the appropriate class:

```
ex: JavaLangClaim an ex:ForeignLanguageClaim;
ex: aboutConcept esco:Java; # Error: esco:Java is a ProgrammingLanguageConcept
ex: cefrLevel "B1".
```

We may also encounter structural errors; one such example is the following, which violates the condition that the initial offset of the EvidenceSpan must be positive:

```
ex: InvalidSpan an ex: EvidenceSpan;
ex: sourceDocument ex:CV1;
ex: startOffset -5; # Error: <0
ex: endOffset 10.
```

By applying SHACL rules, the data extracted from CVs through the NER step is checked against ESCO and filtered locally. Errors are not hidden; they are marked explicitly so they can be used later to improve the classification process. In practice, this makes the system easier to adjust when new types of errors appear. It also handles ambiguous data a bit more reliably. Later, external sources of canonical concepts, such as DBpedia or Wikidata, could also be added.

Table 2. SHACL constraints applied to ontological classes.

Class	Attributes	SHACL Constraints	Error Handling
EvidenceSpan	sourceDocument, startOffset, endOffset, spanConfidence	reference to the source document (sourceDocument required) $\text{startOffset} \geq 0$ endOffset defined spanConfidence in the range [0,1]	If the source-document reference or valid offsets are missing, the associated claim is marked as a critical violation and is excluded from the training-eligible representation.
ProficiencyLevel	label, numericValue	Required label with a numeric value in the range [0,1]	Non-numeric values or values outside the range are rejected
SkillClaim	aboutConcept, hasEvidence, claimedLevel, confidence	Reference to a concept (aboutConcept required) At least one link to EvidenceSpan (hasEvidence exists) confidence in the range [0,1] (optional) claimedLevel required (exactly one ProficiencyLevel)	The absence of aboutConcept or hasEvidence results in the rejection of the statement. If claimedLevel is missing or invalid, the statement is assigned a default value.
ForeignLanguageClaim	aboutConcept, cefrLevel	aboutMandatory concept (exactly one ForeignLanguageConcept) cefrLevel is from the list of levels ["A1", "A2", "B1", "B2", "C1", "C2"]	If aboutConcept is not a foreign language or cefrLevel is not in the defined list, then the statement is marked as non-compliant and receives a predetermined value
ProgrammingLanguageClaim	aboutConcept, hasEvidence	aboutRequired concept (exactly one ProgrammingLanguageConcept) At least one link to EvidenceSpan (hasEvidence exists)	If aboutConcept is not a programming language, the statement is marked as cross-class and is assigned a fixed value
ESCO-Aligned Concept	aboutConcept	aboutConcept must be a valid ESCO URI	If the URI is invalid, the statement is marked as unvalidated and treated as a local URI
LocalUnverifiedConcept	isUnverified	isUnverified = true	These statements are retained in the ontology but explicitly marked as unvalidated and used only in classification contexts based on their similarity score to the canonical ESCO form via fuzzy matching

5. Design and Implementation of the Orchestrated Microservices Architecture for Recruitment Support

An event-based architecture was implemented for the proposed recruitment support system. The system is coordinated by a Spring Boot component that sends messages to Python services. Each service handles a specific part of the recruitment flow, such as entity extraction, structuring, ontology validation, or classification. The architecture is grouped into five main layers as shown in Figure 4:

- Angular 21 frontend, used to build the interface through which users (both candidates and recruiters) interact with the system;
- Java 21 with Spring Boot 3.4.4 backend, acting as the central orchestrator;
- Python workers, responsible for running the AI-related steps including entity extraction, CV classification, and ontology validation;
- RabbitMQ 4.2.5, used as an asynchronous messaging mechanism between the Spring backend and the Python workers;
- PostgreSQL 17, used for storing the data, as well as keeping track of the execution steps for traceability.

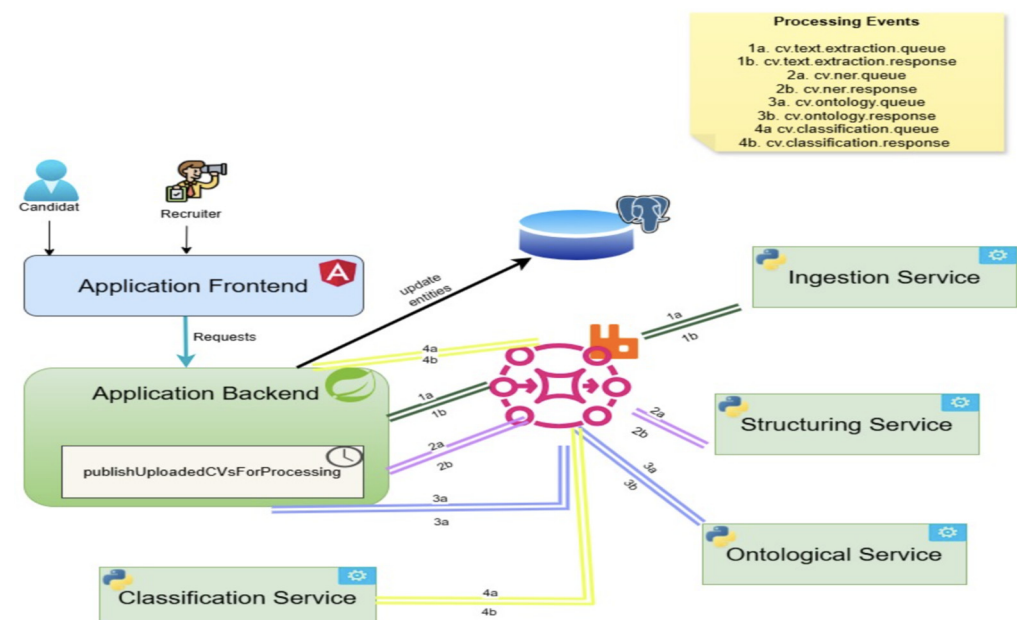


Figure 4. Microservices architecture.

This type of event-based microservices architecture gives better control over handling the execution flow and tracing each step. It also brings together the strengths of the Java ecosystem, mainly for orchestration, security, and transaction control, with the maturity of the Python ecosystem for AI-related tasks.

Similar advantages are also discussed in the literature when it comes to microservices-based workflows. In most cases, the focus is on modularity, scalability, and the ability of software systems to evolve over time in DevOps environments [29,30].

In this setup, the Spring Boot backend is used as the component that coordinates the flow, instead of letting workers communicate directly with each other. This logic is implemented in the `CvPipelineOrchestratorService` class which sends RabbitMQ messages for each stage such as `StructuringRequestMessage`, `OntologyValidationRequestMessage` and `ClassificationRequestMessage`. Each message includes the data needed for that step. In this setup, the orchestrator does more than just trigger execution; it also controls what context is passed to each worker.

The use of RabbitMQ also improves scalability and fault isolation. Each processing stage communicates through explicit queues and routing keys, so the structuring, ontology validation, and classification workers can be scaled independently. If the ontology service is slower than the NER service, additional ontology workers can be deployed without changing the other services. If a Python worker fails, the failure remains localized to that processing stage, while the Spring Boot orchestrator can retry, log, or mark the corresponding task as failed. This decoupling is important because NLP extraction and semantic reasoning have different computational costs and may evolve independently.

As shown in Figures 4 and 5, the processing flow is coordinated by the Spring Boot backend through the CvPipelineOrchestratorService class. Its role is to control how each Python worker is executed by sending specific messages through the RabbitMQ broker. Processing begins with the publishUploadedCVsForProcessing method, which sends a TextExtractionRequestMessage containing the candidate ID and the path to the uploaded CV. This message is handled by the ingestion service. It extracts the raw text using PyMuPDF and saves it in the processed_text column of the Candidate_CVS table. After that, the result is sent back through the cv.text.extraction.response message.

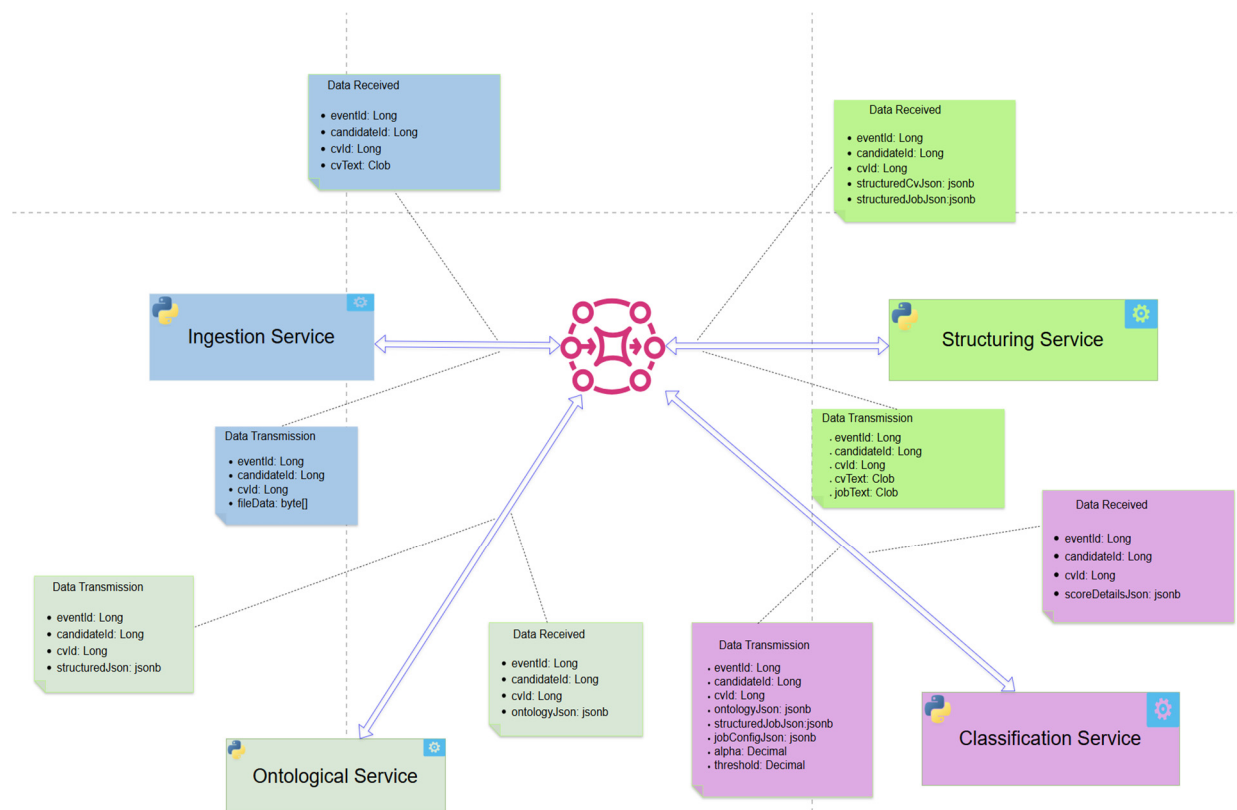


Figure 5. Data transmitted within microservices. Arrows indicate communication between the microservices and RabbitMQ, whereas dashed lines link each microservice to the data structures it transmits and receives.

In the next step, the orchestrator sends a StructuringRequestMessage that contains the pre-processed text. This message is handled by the Structuring service which uses a fine-tuned BERT model for entity recognition. The output is a structured JSON built around the six semantic categories that cover the areas: programming languages, foreign languages, certifications, work experience, soft skills, and education. It also keeps basic position information from the original text. The result is sent back through the cv.ner.response message and stored in the structured_json column.

After that, the backend sends an `OntologyValidationRequestMessage` with the structured JSON to the ontology microservice. This service transforms the data into RDF instances aligned with the defined ontology. It also performs mapping to canonical concepts including integration with ESCO and applies SHACL rules for semantic validation. The output is a semantically validated JSON, together with information about possible inconsistencies. This is returned through the `cv.ontology.response` message and stored in the `ontology_result` column.

In the final stage, the backend sends a `ClassificationRequestMessage` with the validated data to the classification service. This service uses an XGBoost model to compute the candidate–job matching score. The result contains the final score together with a breakdown by category. The result goes into the `CV_Classification_Results` table and is also sent back through the `cv.classification.response` message. In this step, the model links the extracted categories to common IT roles. These include roles like Data Scientist, Machine Learning Engineer, DevOps Engineer, Software Developer, QA Engineer, Project Manager or Cloud Architect. The weights used in scoring can be adjusted by recruiters if the job requirements change.

This asynchronous request–response communication model makes it easier to track what happens between services. Each step is tied to a specific message, and intermediate results are stored in the database. This helps with both traceability and monitoring of the entire pipeline.

For scaling and decoupling, RabbitMQ is used as the integration layer between the Spring backend and the Python services. Communication uses a `cv.pipeline.exchange` defined in Java with separate routing keys and queues for each stage, such as entity extraction, ontology validation, and classification. Based on the RabbitMQ documentation, durable queues and persistent messages help keep data safe even if the broker restarts. At the same time, acknowledgement mechanisms and publisher confirms ensure that messages are delivered correctly on both the consumer and producer sides [31].

Message publishing is handled through the `RabbitTemplate` class provided by Spring AMQP. This is used to send messages to the `cv.pipeline.exchange` using the routing keys defined for each stage. In this way, the system can control how many tasks run in parallel and avoid overloading the services, especially in cases where expensive operations are involved such as NLP used to build structured JSON data from CV parsing [32].

The architecture also supports a consistent way of handling data by using the `jsonb` type in PostgreSQL. According to the official documentation, this allows efficient storage, indexing, and querying of JSON data within a distributed processing flow [33]. In practice, the outputs from the three main microservices entity extraction, ontology validation, and classification end up in the same place, so they can be used together without extra steps. Data access remains efficient.

PostgreSQL is used as the main database. Results from ingestion, structuring, and ontology validation are saved in the `Candidate_CVS` table in the fields `processed_text`, `structured_json` and `ontology_result`. The `processing_status` is updated step by step with values such as `TEXT_EXTRACTED`, `STRUCTURED` and `ONTOLOGY_VALIDATED`. The table also reserves fields for optional application-level parameters and stores `model_version` and `score_details_json`. These optional parameters were not used in the training or benchmark evaluation reported in this study. The `score_details_json` field records how the semantic dimensions and ontology-derived quality indicators contributed to the resulting classification. The latter is stored as `jsonb` and provides details about how each semantic category contributed to the final score. This helps make the decision process more transparent. In this setup, PostgreSQL becomes a central part of the architecture, both for data storage and for tracking the processing flow.

Reliability is handled at different levels in the system. On the infrastructure side, RabbitMQ helps limit data loss by using durable queues, persistent messages, and acknowledgement mechanisms. At the application level, traceability is supported by storing the results of each intermediate step in PostgreSQL. The execution itself is controlled by the Spring Boot orchestrator, which provides a stable model for building distributed applications, easy integration with messaging systems through Spring AMQP, and support for asynchronous programming [34].

Another important aspect is the clear separation between the orchestrator and the Python workers. Communication happens only through messages. This makes it possible to scale the system horizontally, extend it by adding new types of workers, and isolating failures. These characteristics follow the general principles of microservices and event-driven architecture [32].

The additional semantic processing introduced by the proposed architecture comes with a computational cost. Compared with approaches based solely on keyword matching or direct text comparison, the workflow includes several extra stages, such as document extraction, transformer-based entity recognition, ontology population, concept alignment, and SHACL validation. Each of these operations contributes to the overall processing time.

The impact becomes more noticeable when large numbers of CVs are submitted within a short period of time or when concept verification requires interaction with external knowledge sources. Under these conditions, the semantic validation stage may introduce additional latency compared with lightweight recruitment systems that rely only on lexical matching.

To reduce this effect, the architecture was designed around asynchronous communication and independent processing services. Tasks are distributed through message queues, allowing different stages of the pipeline to execute separately and at their own pace. Since the Python workers responsible for extraction, validation, and classification operate independently, additional instances can be deployed when higher workloads are expected.

Even with these optimizations, the proposed solution prioritizes semantic quality and traceability over minimal response time. For this reason, it is particularly suitable for recruitment workflows involving controlled candidate screening, staged evaluation processes, or batch processing of applications. Scenarios that require immediate ranking of large numbers of CVs may require additional optimizations, such as caching strategies or precomputed semantic representations.

6. Results and Experimental Evaluation

This section evaluates the main components of the proposed pipeline. Since the classification model depends on the quality of the structured CV representation, the NER model is first evaluated separately. The following subsections then report the effect of ontology-based validation on the CV–job classification task.

Table 3 summarizes the relationship among the datasets, model configurations, and evaluation stages. The NER experiment evaluates information extraction independently, whereas the classification experiments compare the original corpus, size-matched random controls, and two models trained on the same ontology-filtered corpus. The external benchmark is used only for final evaluation and does not contribute to training, probability calibration, or threshold selection.

Table 3. Summary of the datasets, model configurations, and evaluation purposes.

Configuration	Data and Size	Evaluation Procedure	Purpose
NER model	11,802 annotated resume documents	Document-level 80/10/10 split	Independent evaluation of entity extraction
V1-Full	12,000 CVs, 200 job descriptions, 120,000 pairs	Double-disjoint split	Full-corpus baseline
V1-Random-35770	Ten size-matched samples containing 35,770 pairs	Double-disjoint split for every seed	Controls for dataset-size reduction
V2-Filtered-NoOntologyFeatures	3577 retained CVs, 196 jobs, 35,770 pairs	Exact same pairs and split as V2-Ontology-35770	Measures filtered-corpus performance without validation-derived features
V2-Ontology-35770	The same 35,770 filtered pairs	Same split, calibration, and threshold-selection procedure	Measures the contribution of ontology-derived quality features
External benchmark	2520 pairs, 63 candidates, 125 job descriptions	Evaluation only	Tests transfer to an independently labelled corpus

6.1. NER Training Setup

The document structuring service relies on a fine-tuned xlm-roberta-base model used for named entity recognition. Its purpose is to identify the information needed later by the ontology and classification services and convert the raw CV text into a structured representation. The extracted entities are then mapped to ontology concepts and used as input features during classification.

The training data was assembled from two publicly available resume annotation datasets: Mehyaar/Annotated_NER_PDF_Resumes (https://huggingface.co/datasets/Mehyaar/Annotated_NER_PDF_Resumes, accessed on 15 June 2026) and DataTurks Resume Entities for NER (<https://www.kaggle.com/datasets/dataturks/resume-entities-for-ner>, accessed on 15 June 2026). The first dataset focuses mainly on technical skills found in IT resumes, while the second contains annotations for more general resume-related information. Because the two sources follow different annotation schemes, their labels were first aligned with the semantic categories used throughout our system.

To obtain a unified corpus, the original annotations were complemented with domain dictionaries and a set of rule-based procedures. The dictionaries covered programming languages and technologies, foreign languages, certifications, soft skills, educational terms, and proficiency levels as shown in Table 4. In addition, regular expressions were used to identify experience durations, common IT job titles, certification names, and CEFR language levels ranging from A1 to C2. This combination of manual resources and deterministic rules helped produce annotations that match the semantic structure expected by the ontology and classification components.

The final dataset follows the BIO tagging format commonly used in NER tasks (An example of the BIO annotation scheme is presented in Table 5). Each entity is represented through beginning (B) and inside (I) tags, while tokens that do not belong to any entity are assigned the label O. In total, the dataset contains 17 labels, including CERTIFICATION, EDUCATION, EXPERIENCE_YEARS, FOREIGN_LANGUAGE, LEVEL, PROGRAMMING_LANGUAGE, SOFT_SKILL, and WORK_EXPERIENCE together with their corresponding BIO variants.

Table 4. Entity labels used for NER fine-tuning.

Entity Type	Meaning in the Proposed System	Example Span
PROGRAMMING_LANGUAGE	Programming languages, frameworks, tools, and technical skills	Java, Spring Boot, SQL, Docker
FOREIGN_LANGUAGE	Natural languages declared by the candidate	English, French, German
LEVEL	Proficiency level for programming or foreign languages	advanced, intermediate, B2, C1
CERTIFICATION	Professional or technical certifications	Certified Kubernetes Administrator, AWS Certified Solutions Architect
WORK_EXPERIENCE	Professional role or job title	Java Developer, Data Analyst, DevOps Engineer
EXPERIENCE_YEARS	Duration of professional experience	5 years of experience, 2020 to present
SOFT_SKILL	Non-technical skills relevant to recruitment	communication, teamwork, leadership
EDUCATION	Education-related information	Bachelor’s degree in computer science

Table 5. Example of BIO annotation.

Token	BIO Label
English	B-FOREIGN_LANGUAGE
B2	B-LEVEL
Advanced	B-LEVEL
Java	B-PROGRAMMING_LANGUAGE
5	B-EXPERIENCE_YEARS
years	I-EXPERIENCE_YEARS
of	I-EXPERIENCE_YEARS
experience	I-EXPERIENCE_YEARS

The final corpus contained 11,802 annotated resume documents, split at document level into 9441 training documents, 1180 validation documents, and 1181 test documents, corresponding approximately to an 80/10/10 split. This split was used only for evaluating the NER component. Table 6 summarizes the hyperparameters used for fine-tuning.

Table 6. NER fine-tuning hyperparameters.

Parameter	Value
Learning rate	1×10^{-5}
Training epochs	6
Train batch size per device	4
Evaluation batch size per device	4
Gradient accumulation steps	2
Weight decay	0.01
Warm-up ratio	0.1
Evaluation strategy	At the end of each epoch
Save strategy	At the end of each epoch
Best model selection metric	macro_f1
Mixed precision	fp16 = True
Logging steps	50

6.2. NER Entity-Level Results

The evaluation was conducted using data that was not included in the training. Precision, recall, and F1-score were calculated at entity level. The resulting scores were 0.822 micro-F1, 0.768 macro-F1, and 0.824 weighted-F1. Precision was measured at 0.772 and recall at 0.879.

The distribution of scores across entity categories was not uniform. Entities related to foreign languages, proficiency levels and experience duration were generally recognized without major difficulties as shown in Table 7. These entities are often present in a consistent format across various CVs. It is important to note that similar wording is often used regardless of the candidate, which makes it easier to detect. As demonstrated by the results obtained for FOREIGN_LANGUAGE, LEVEL, and EXPERIENCE_YEARS, this is evident.

Table 7. Entity-level NER performance.

Entity Type	Precision	Recall	F1-Score	Support
CERTIFICATION	0.59	0.42	0.49	115
EDUCATION	0.69	0.82	0.75	1259
EXPERIENCE_YEARS	0.87	0.92	0.89	1143
FOREIGN_LANGUAGE	0.91	0.99	0.95	681
LEVEL	0.89	0.94	0.91	1037
PROGRAMMING_LANGUAGE	0.76	0.87	0.81	2162
SOFT_SKILL	0.74	0.89	0.81	1715
WORK_EXPERIENCE	0.46	0.66	0.54	224
Micro avg.	0.77	0.88	0.82	8336
Macro avg.	0.74	0.81	0.77	8336
Weighted avg.	0.78	0.88	0.82	8336

Different behaviours were observed for certifications and work experience entries. During manual inspection of the predictions, several cases were identified where the boundaries of the extracted entities were not clearly defined. Certification names often contain references to technologies, platforms, or professional roles. Likewise, job titles frequently include technical terms that may also appear in certification titles or skill descriptions. An example encountered during evaluation was the expression “Certified Kubernetes Administrator”, where the distinction between certification-related information and role-related information is not always straightforward.

These cases illustrate a limitation that cannot always be addressed through statistical learning alone. Even when an entity is detected correctly, its semantic interpretation may remain ambiguous. For this reason, the extracted entities are subsequently processed by the ontology service, where additional semantic checks are performed. The validation stage verifies whether the extracted concept is compatible with the assigned category and whether it can be linked to a valid ontology concept.

From a practical perspective, the evaluation indicates that the NER component provides sufficiently accurate extraction for most of the CV content used later in the recruitment pipeline. At the same time, the analysis highlights a few situations where semantic validation remains necessary before the extracted information is used by the CV-job matching model.

6.3. Ontology-Guided Dataset Refinement and XGBoost Classification

The second part of the evaluation concerns the CV-job classification stage. The purpose of this experiment was to determine whether ontology-based validation improves the quality and robustness of the feature representation used by the XGBoost classifier.

The training data originates from two distinct sources: the Resume-Atlas CV (<https://huggingface.co/datasets/ahmedheakl/resume-atlas>, accessed on 22 June 2026) database and the Kaggle Jobs and Job Description dataset (<https://www.kaggle.com/datasets/kshitizregmi/jobs-and-job-description>, accessed on 22 June 2026). The choice of these datasets was motivated by the fact that both contain information that can be associated at the occupational level within the IT recruitment domain. While the first source describes the candidate side of the matching problem, the second provides the corresponding representation of job requirements. To structure the feature space, the processing service extracts six semantic dimensions from each CV. These semantic fields cover programming languages, soft skills, certifications, foreign languages, professional experience, and education. Job descriptions undergo the same semantic transformation. Consequently, each CV–job pair is mapped into a shared feature space for comparison.

The first dataset version (V1) was built by combining 12,000 CVs with a collection of 200 job descriptions. Generating all possible CV–job combinations would have produced an excessively large corpus and a highly imbalanced matching space. Instead, a controlled sampling strategy was adopted. For each CV, a fixed random seed (42) was used to select 10 job descriptions from the available pool. This procedure resulted in 120,000 CV–job pairs while avoiding the construction of a complete Cartesian product. A weak-supervision procedure generates these labels, quantifying semantic compatibility between candidate profiles and job descriptions. The underlying configuration fixes a 0.40 target positive ratio, a 0.11 semantic compatibility threshold, and a 0.06 boundary margin to exclude borderline uncertainties. Filtration yields 53,361 positive and 66,639 negative pairs, anchoring the empirical positive ratio at 44.47% as reported in Table 8.

Table 8. Dataset construction and ontology-feature use in the classification experiments.

Variant	CVs Used for Pair Construction	Job-Description Pool After Filtering	Jobs Assigned per CV	CV–Job Pairs	Label 0	Label 1	Positive Ratio (%)	Ontology Derived Quality Features
V1-Full	12,000	200	10	120,000	66,639	53,361	44.47	No
V1-Random-35770	3577	200	10	35,770	19,868.1 ± 106.1	15,901.9 ± 106.1	44.46 ± 0.3	No
V2-Filtered-NoOntologyFeatures	3577	196	10	35,770	20,035	15,735	43.99	No
V2-Ontology-35770	3577	196	10	35,770	20,035	15,735	43.99	Yes

The generated label does not represent a simple similarity measure. Instead, it reflects the degree of semantic compatibility between a CV and a job description. The computation considers role compatibility together with information extracted from the six semantic dimensions considered throughout the system. Matching therefore depends on the overlap between entities, agreement between occupational categories, and the correspondence between declared and required competency levels.

Not all dimensions contribute in the same way. Certifications, education and soft skills are evaluated through overlap-based measures, whereas programming languages, foreign languages, and professional experience also consider proficiency levels. In place of qualitative proficiency scales, numerical baselines are used through fixed assignments: beginner equals 0.25, intermediate reaches 0.50, advanced marks 0.75, and expert hits the 1.00 ceiling. Consider an intermediate Java claim 0.50 matched against an advanced job mandate 0.75. This configuration requires a direct division to fix the final level ratio at 0.66.

The second dataset version (V2) introduces the ontology validation stage. Instead of relying only on the structured JSON produced by the NER service, the extracted en-

tities are processed by the ontology and semantic validation service. During this stage, entities are transformed into RDF/OWL instances, linked to textual evidence through EvidenceSpan, associated with normalized proficiency levels through ProficiencyLevel, and validated using SHACL constraints. ESCO is used as the reference vocabulary for canonical concept alignment.

Therefore, the classification model receives not only semantic features but also information describing the quality of those features. Examples include conformance status, training eligibility, recoverable non-conformance, violation counts, semantic confidence, and the proportion of concepts that could not be verified against the ontology.

A profile is considered training-eligible when the extracted information remains traceable to the original document and the detected inconsistencies can still be corrected. Typical examples include cross-class assignments resolved through ontology mapping, ambiguous competency levels normalized after extraction, or concepts that cannot be fully verified but are explicitly marked as uncertain. Profiles lacking textual evidence, containing invalid offsets, or failing semantic validation are excluded from the training process.

After ontology filtering, 3577 CVs remained from the original collection. A similar validation process reduced the job description set from 200 to 196 instances. Using the same pairing strategy adopted in V1, each retained CV was associated with 10 job descriptions, producing a final corpus of 35,770 CV-job pairs.

A separate control dataset was created to distinguish the effect of ontology validation from the effect of dataset reduction alone. Ten different random seeds (from one to 10) were used. For each seed, 3577 candidates were sampled from the original collection, while preserving the same pairing strategy of 10 job descriptions per candidate. This produced 35,770 pairs for every sampled dataset, exactly matching the size of the ontology-enhanced corpus. The sampling procedure was applied at candidate level rather than at pair level. Maintaining this constraint ensured that the control datasets preserved the same structural characteristics as the ontology-filtered dataset, making the subsequent comparison more meaningful.

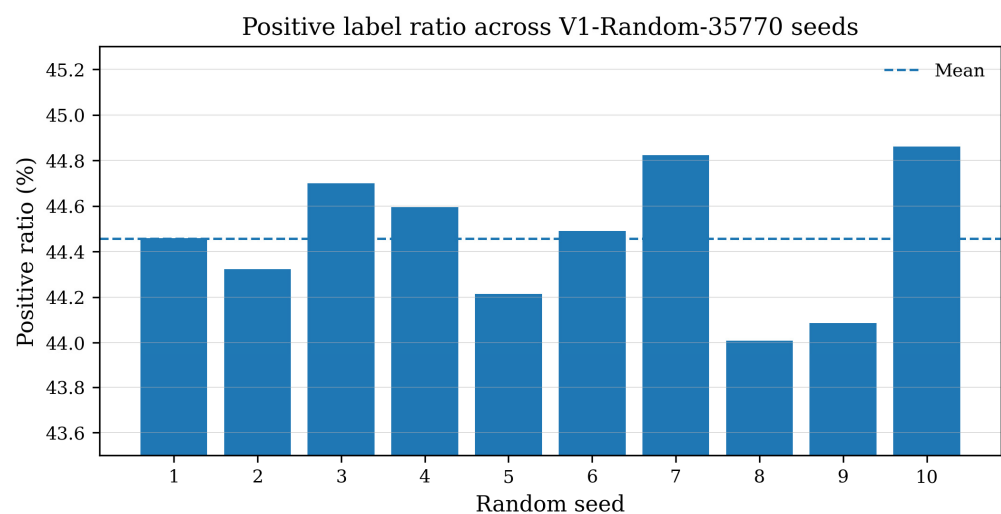
The size-matched random controls determine whether arbitrary dataset reduction can explain the behaviour of V2-Ontology-35770. However, they do not distinguish the effect of selecting semantically cleaner profiles from the contribution of the ontology-derived quality features, because both aspects differ between V1-Random-35770 and V2-Ontology-35770.

To address this remaining confounding factor, an additional configuration, denoted V2-Filtered-NoOntologyFeatures, was trained on the exact same 35,770 ontology-filtered CV-job pairs used by V2-Ontology-35770. The candidate identifiers, job-description identifiers, pair identifiers, class labels, double-disjoint partition, semantic matching features, probability-calibration procedure, and threshold-selection strategy were unchanged. The only difference was that V2-Filtered-NoOntologyFeatures excluded the ontology-derived quality indicators: conformance status, training eligibility, recoverable non-conformance, violation counts, unverified concept ratio, and semantic confidence. Therefore, the direct comparison between the two filtered configurations isolates the contribution of the validation-derived feature set while keeping profile selection and dataset construction constant.

The size-matched random controls were generated using seeds 1 to 10. Their label distributions are reported in Table 9 and Figure 6. The positive ratio remains close to that of the full V1 dataset, indicating that the random control preserves the class balance of the original training data. Therefore, differences between V1-Random-35770 and V2-Ontology-35770 cannot be attributed simply to a major shift in positive/negative class distribution.

Table 9. Label distribution of the V1-Random-35770 size-matched controls.

Seed	Pairs	Label 0	Label 1	Positive Ratio (%)
1	35,770	19,867	15,903	44.46
2	35,770	19,916	15,854	44.32
3	35,770	19,781	15,989	44.70
4	35,770	19,818	15,952	44.60
5	35,770	19,955	15,815	44.21
6	35,770	19,856	15,914	44.49
7	35,770	19,737	16,033	44.82
8	35,770	20,028	15,742	44.01
9	35,770	20,000	15,770	44.09
10	35,770	19,723	16,047	44.86
Mean $\pm$ SD	35,770	19,868.1 $\pm$ 106.1	15,901.9 $\pm$ 106.1	44.46 $\pm$ 0.30

**Figure 6.** Positive label distribution of random seeds.

To prevent information leakage, all configurations used a double-disjoint split. In recruitment datasets, the same candidate profile or the same job description may appear in multiple pairs, which can lead to overly optimistic results if the split is performed directly at pair level. To avoid this situation, candidates and job descriptions were partitioned independently using a fixed random seed. A CV–job pair was included in the training set only when both the candidate and the corresponding job description belonged to the training partition. The same rule was applied to the internal test set. Pairs that combined elements from different partitions were removed from the experiment.

Consequently, no candidate profile and no job description could appear in both training and testing. This creates a stricter evaluation setting, since the model is required to make predictions for candidates and jobs that were not encountered during training.

The feature representation used by the classification model was built at two different levels of detail. On the one hand, it captured the similarity between broader semantic categories. On the other hand, it preserved information about the individual entities extracted from CVs and job descriptions. For each semantic category, only the most relevant item-level matches were retained, using a top-k strategy with k equal to 6. The categories considered during training were programming languages, soft skills, certifications, foreign languages, professional experience, and education.

An earlier version of the feature set also included a structural cosine similarity score intended to provide a global comparison between candidate and job profiles. During preliminary experiments, however, this feature tended to dominate the decision process.

For this reason, it was removed from the final representation to encourage the model to rely on the individual semantic dimensions rather than on a single aggregate similarity measure.

The classification stage was implemented using XGBoost, followed by a sigmoid calibration step to obtain calibrated probability estimates. Model parameters were selected through a randomized search procedure combined with five-fold stratified cross-validation on the training data. Because the number of positive and negative pairs was not perfectly balanced, class weights were adjusted through the `scale_pos_weight` parameter.

To ensure a fair comparison between dataset versions, the same classification framework was used throughout all experiments. The feature generation process, calibration procedure, train-test split strategy, and model family remained unchanged for V1-Full, V1-Random-35770, and V2-Ontology-35770. The same procedure was used for V2-Filtered-NoOntologyFeatures. Within the direct filtered corpus ablation, only the ontology-derived feature columns were removed; all other experimental conditions remained unchanged.

The selected hyperparameters were also largely identical across configurations, except for `scale_pos_weight`, whose value depends directly on the class distribution observed in the corresponding training partition as reported in Table 10.

Table 10. XGBoost configuration and selected hyperparameters for the evaluated classification variants.

Parameter	V1-Full	V2-Filtered-NoOntologyFeatures	V2-Ontology-35770
n_estimators	600	600	600
max_depth	5	5	5
learning_rate	0.040842	0.040842	0.040842
subsample	0.95	0.95	0.95
colsample_bytree	0.60	0.60	0.60
min_child_weight	2	2	2
gamma	2.00	2.00	2.00
reg_alpha	0.0001	0.0001	0.0001
reg_lambda	10.00	10.00	10.00
scale_pos_weight	1.8773	1.9508	1.9508
Tree method	hist	Hist	hist
Probability calibration	sigmoid	Sigmoid	sigmoid
Calibration subset	12% of training partition	12% of training partition	12% of training partition
Threshold-validation subset	12% of training partition	12% of training partition	12% of training partition
Best CV ROC-AUC	0.8659	0.8893	0.8939

The decision threshold was not selected on the external benchmark. The training partition was split into three parts: a model-training subset, a calibration subset, and a threshold-validation subset. The threshold used for the main evaluation was selected on the threshold-validation subset by maximizing F1-score and was then applied unchanged to the internal test set and to the external benchmark. This prevents the benchmark from being used for model selection. Benchmark-best thresholds are reported only for threshold-sensitivity analysis.

The V1-Random-35770 controls used the same model family, search space, calibration procedure, and split strategy. Class-dependent parameters, including `scale_pos_weight`, were calculated separately for each sampled training partition.

The external benchmark used in the evaluation was derived from the publicly available Hugging Face dataset `facehuggerapoorv/resume-jd-match` (<https://huggingface.co/datasets/facehuggerapoorv/resume-jd-match/>, accessed on 24 June 2026). The dataset con-

tains pairs of resume texts and job-description texts together with a binary label indicating whether the candidate profile matches the corresponding job description.

The benchmark was used exclusively for external evaluation. None of its samples were involved in model training, hyperparameter optimization, probability calibration, or threshold selection. This separation was maintained to provide an independent assessment of model behaviour outside the data used during development.

Because the focus of the present study is recruitment within the IT domain, an additional filtering step was applied before the benchmark was used. Job descriptions unrelated to information technology were removed, and only IT-oriented positions were retained. The objective of this filtering stage was to ensure that the evaluation data remained consistent with the occupational profiles considered throughout the training and ontology validation process.

After applying the domain-specific filtering criteria, the final benchmark consisted of 2520 CV–job pairs. It included 63 unique candidates and 125 unique job descriptions as reported in Table 11. Each candidate was associated with 40 benchmark job-pairing instances. The benchmark label distribution contained 1499 negative pairs and 1021 positive pairs, corresponding to a positive ratio of 40.52%. The ground-truth labels were taken from the original dataset labels and were not generated using the semantic-compatibility rule, cosine similarity, or the XGBoost classifier. This separation is important because the benchmark is independent from the Resume-Atlas and Kaggle Jobs and Job Description sources used to construct the training data. The same benchmark file was used for V1-Full, all V1-Random-35770 seeds, V2-Filtered-NoOntologyFeatures, and V2-Ontology-35770, allowing paired comparison and McNemar’s test.

Table 11. External benchmark distribution.

Benchmark	Candidates	Unique Job Descriptions	Job-Pairing Instances per Candidate	Pairs	Label 0	Label 1	Positive Ratio (%)
Common benchmark	63	125	40	2520	1499	1021	40.52

The main results are reported in Table 12 and illustrated in Figure 7. All values are computed at the validation-selected threshold. For V1-Random-35770, results are reported as mean $\pm$ standard deviation across the 10 seeds.

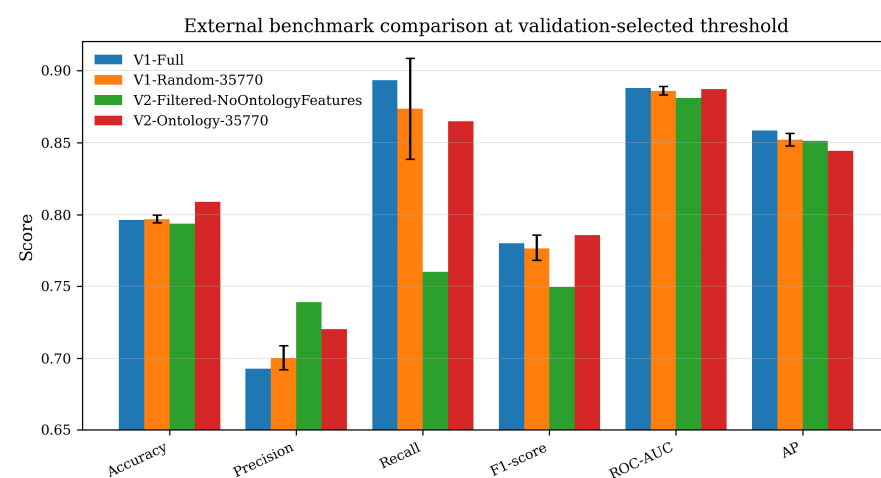


Figure 7. External benchmark comparison of V1-Full, V1-Random-35770, V2-Filtered-NoOntologyFeatures, and V2-Ontology-35770 at the validation-selected threshold.

Table 12. Internal and external benchmark results at the validation-selected threshold.

Variant	Threshold	Internal Acc.	Internal Prec.	Internal Rec.	Internal F1	Internal ROC-AUC	Internal AP	Benchmark Acc.	Benchmark Prec.	Benchmark Rec.	Benchmark F1	Benchmark ROC-AUC	Benchmark AP
V1-Full	0.22	0.733	0.633	0.950	0.760	0.867	0.843	0.796	0.693	0.893	0.780	0.888	0.859
V1-Random-35770	0.263 ± 0.044	0.742 ± 0.014	0.645 ± 0.026	0.918 ± 0.027	0.757 ± 0.015	0.862 ± 0.009	0.834 ± 0.018	0.797 ± 0.003	0.700 ± 0.008	0.873 ± 0.035	0.777 ± 0.009	0.886 ± 0.003	0.852 ± 0.004
V2-Filtered-NoOntology-Features	0.32	0.785	0.702	0.927	0.799	0.893	0.876	0.794	0.739	0.760	0.749	0.881	0.851
V2-Ontology-35770	0.28	0.781	0.699	0.927	0.797	0.893	0.879	0.809	0.720	0.865	0.786	0.887	0.844

The internal results of the two filtered-corpus configurations are very similar. V2-Filtered-NoOntologyFeatures achieves an internal accuracy of 0.785, precision of 0.702, recall of 0.927, and F1-score of 0.799, whereas V2-Ontology-35770 obtains 0.781, 0.699, 0.927, and 0.797, respectively. Their ROC-AUC values are also nearly identical at 0.893. These results do not indicate a clear internal advantage for the ontology-derived quality features. Since the internal labels are generated through weak supervision from the same semantic dimensions later used by the classifier, the internal evaluation is interpreted mainly as a consistency check within the constructed data distribution rather than as independent evidence of generalization.

The external benchmark provides a clearer distinction between the two filtered configurations. V2-Filtered-NoOntologyFeatures and V2-Ontology-35770 were trained on the exact same 35,770 CV-job pairs, using the same candidates, job descriptions, class labels, double-disjoint split, calibration procedure, and threshold-selection strategy. The inclusion of ontology-derived quality indicators marks the sole divergence for V2-Ontology-35770, isolating the baseline feature contribution from any training corpus cleaning effects. This modification shifts benchmark accuracy to 0.809 (from 0.794) and recall to 0.865 (from 0.760), while F1-score and ROC-AUC climb to 0.786 and 0.887, respectively.

The comparison with V1-Full and the size-matched random controls addresses a different question. V2-Ontology-35770 obtains an accuracy of 0.809, compared with 0.796 for V1-Full and 0.797 ± 0.003 for V1-Random-35770. Its precision is 0.720, compared with 0.693 and 0.700 ± 0.008 , while its F1-score is 0.786, compared with 0.780 and 0.777 ± 0.009 . These results indicate that arbitrary dataset-size reduction does not reproduce the same selected-threshold behaviour. However, the random-control comparison alone cannot isolate the contribution of the ontology-derived features because it also changes the candidates retained for training. For this reason, the direct comparison with V2-Filtered-NoOntologyFeatures provides the stronger feature-level ablation.

The impact of ontology validation diverges across evaluation metrics. When contrasted with V1-Full and the size-matched random controls, V2-Ontology-35770 exhibits a minor reduction in benchmark recall, sliding to 0.865 (from 0.893 and 0.873 ± 0.035 , respectively). Average Precision undergoes an identical contraction, dropping to 0.844 from baseline markers of 0.859 and 0.852 ± 0.004 . Isolating the feature-level effect through the filtered-dataset control where training sample distributions are identical reverses this dynamic for key metrics. Within this controlled architecture, the ontology-derived quality features drive benchmark recalls up to 0.865 (from 0.760), compressing false negatives from 245 down to 138. Conversely, precision drops slightly from 0.739 to 0.720, Average Precision settles at 0.844 (down from 0.851), and false positives scale from 274 to 343.

These observations are easier to interpret when considering what each metric measures. Accuracy, precision, and F1-score are computed for a single operating threshold, whereas Average Precision summarizes model behaviour over the entire precision–recall curve. As a result, improvements observed at one decision threshold do not necessarily imply better ranking performance across all possible thresholds.

The ontology-derived quality indicators do not simply make the classifier more conservative or more permissive. Instead, they modify how positive predictions are produced at the operating threshold selected during validation. In the filtered-dataset comparison, this leads to more correctly identified positive matches and substantially fewer false negatives, but it also introduces additional false-positive predictions.

For this reason, the contribution of ontology validation should not be interpreted as a uniform improvement across every evaluation measure. A more accurate interpretation is that the ontology-derived quality indicators change the operating characteristics of the classifier in a statistically significant way. They improve several threshold-dependent

measures, including accuracy, recall, F1-score, ROC-AUC, and paired decision correctness on the external benchmark, while precision and Average Precision exhibit a different trend. The main contribution of these experiments is therefore the demonstration that semantic quality information extracted during ontology validation influences classifier behaviour beyond the effect of filtering the training data alone.

The confusion matrices presented in Table 13 and Figure 8 provide a more detailed view of the precision–recall trade-off observed during evaluation. They help explain how the ontology-derived quality indicators influence the final classification decisions.

Table 13. Benchmark confusion matrix comparison at the validation-selected threshold.

Variant	TN	FP	FN	TP	Interpretation
V1-Full	1095	404	109	912	Highest recall, but more false positives
V1-Random-35770	1116.9 ± 29.8	382.1 ± 29.8	129.2 ± 35.7	891.8 ± 35.7	High recall, moderate false-positive rate
V2-Filtered-NoOntologyFeatures	1225	274	245	776	Higher precision, but a substantial loss of positive matches
V2-Ontology-35770	1156	343	138	883	Recovers substantially more positive matches than the no-feature configuration, with an increase in false positives

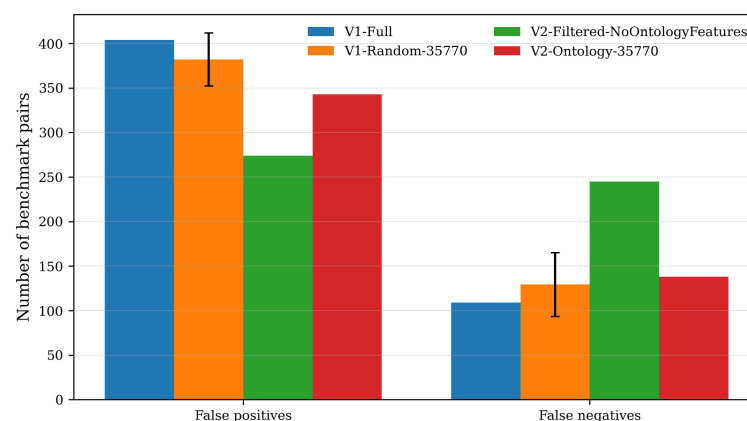


Figure 8. False-positive and false-negative comparison for all evaluated model configurations at the validation-selected threshold. Bars for V1-Random-35770 represents the mean values across the ten random size-matched subsets.

The comparison between the two ontology-filtered configurations is particularly informative because both models are trained on the same dataset. Baseline error rates for V2-Filtered-NoOntologyFeatures stand at 274 false positives and 245 false negatives. The injection of semantic quality indicators halves false negatives to 138 and forces true positives up to 883 (from 776), though at the cost of expanding false positives to 343. This

structural trade-off demonstrates that ontology indicators alter the model's activation thresholds globally, refuting any simple shift toward strictness or permissiveness. Instead, they modify the balance between missed matches and additional positive predictions, shifting the operating point toward higher recovery of suitable candidates.

In institutional recruitment, this shift directly alters the risk profile of the selection funnel. Because elevated false negatives represent an operational cost—specifically, the exclusion of qualified talent—the system's operating threshold must align with the specific screening phase. Consequently, early-stage sourcing pipelines benefit from a lowered threshold to maximize applicant recall, whereas final-stage filtering requires an elevated threshold to secure high-precision shortlists.

The reduction in recall observed after ontology validation motivated a more detailed analysis of the classifier behaviour across different decision thresholds. Figure 9 presents precision, recall, and F1-score measured on the external benchmark for a range of threshold values. The threshold selected on the validation subset was retained for the main evaluation results, while the benchmark threshold sweep was used only to explore alternative operating points.

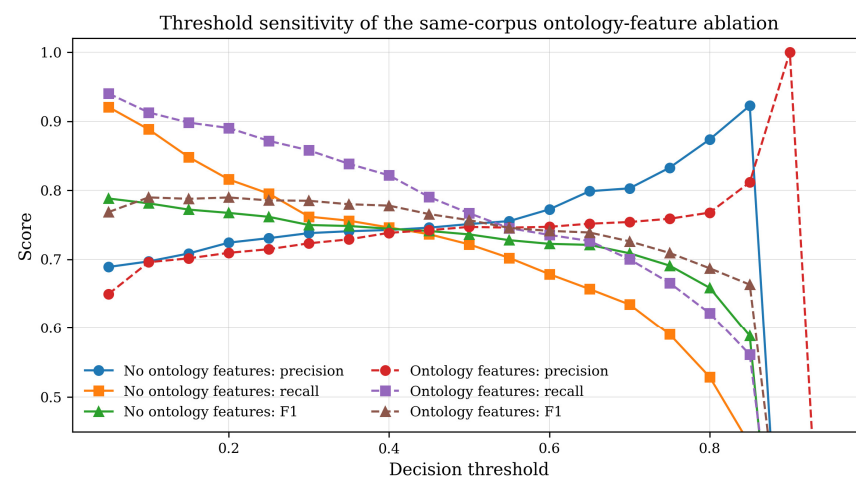


Figure 9. Threshold-sensitivity comparison between V2-Filtered-NoOntologyFeatures and V2-Ontology-35770 on the external benchmark.

The analysis shows that the same model can support different recruitment strategies depending on the desired balance between precision and recall. Lower threshold values tend to increase recall, allowing a larger number of potentially suitable candidates to be retained. Higher threshold values have the opposite effect, reducing the number of positive predictions and increasing precision. From a practical perspective, this makes it possible to adapt the behaviour of the classifier to different recruitment scenarios without modifying the underlying model.

In addition to the threshold analysis, statistical significance was examined using McNemar's exact test. The comparison was performed on the same set of 2520 benchmark CV-job pairs, ensuring that both models were evaluated on identical observations. The paired analysis used the prediction column generated with the validation-selected threshold (`pred_validation_best_f1`), so all comparisons were based on the operating point reported in the main evaluation.

Table 14 summarizes the disagreement counts required by McNemar's test. The value denoted by *b* corresponds to cases where the baseline prediction was correct and the ontology-enhanced model was incorrect. Conversely, variable *c* isolates the instances where the ontology-enhanced model rectifies the baseline's misclassifications. When *c* outpaces *b*, this directional asymmetry demonstrates that the ontological framework cor-

rects significantly more errors than it introduces, providing statistical validation for the proposed approach.

Table 14. McNemar paired comparisons on the common benchmark.

Comparison	Common Pairs	b: Baseline Correct, V2-Ontology- 35770 Wrong	c: Baseline Wrong, V2-Ontology- 35770 Correct	Exact <i>p</i> -Value	Effect Ratio c/b
V2-Ontology-35770 vs. V2-Filtered- NoOntologyFeatures	2520	70	108	0.00540	1.54
V2-Ontology-35770 vs. V1-Full	2520	29	61	0.00097	2.10
V2-Ontology-35770 vs. V1-Random-35770 seed 1	2520	32	62	0.00259	1.94
V2-Ontology-35770 vs. V1-Random-35770 seed 2	2520	29	70	0.00005	2.41
V2-Ontology-35770 vs. V1-Random-35770 seed 3	2520	34	69	0.00073	2.03
V2-Ontology-35770 vs. V1-Random-35770 seed 4	2520	29	62	0.00071	2.14
V2-Ontology-35770 vs. V1-Random-35770 seed 5	2520	34	58	0.01602	1.71
V2-Ontology-35770 vs. V1-Random-35770 seed 6	2520	33	56	0.01919	1.70
V2-Ontology-35770 vs. V1-Random-35770 seed 7	2520	47	70	0.04150	1.49
V2-Ontology-35770 vs. V1-Random-35770 seed 8	2520	45	85	0.00057	1.89
V2-Ontology-35770 vs. V1-Random-35770 seed 9	2520	36	66	0.00387	1.83
V2-Ontology-35770 vs. V1-Random-35770 seed 10	2520	35	59	0.01721	1.69
V2-Ontology-35770 vs. V1-Random-35770 mean	2520	35.4 ± 6.1	65.7 ± 8.5	0.01024 ± 0.01341	1.88 ± 0.27

The primary comparison for isolating the contribution of the ontology-derived quality features is between V2-Filtered-NoOntologyFeatures and V2-Ontology-35770. Both configurations were trained on the same 35770 ontology-filtered CV-job pairs, using identical labels, candidate/job partitions, calibration procedures, and threshold-selection strategies. The

presence of ontology-derived quality indicators forms the sole divergence between these identical-corpus models. McNemar's exact test isolates this structural delta: V2-Ontology-35770 resolves 108 benchmark anomalies induced by V2-Filtered-NoOntologyFeatures, while the baseline architecture corrects only 70 opposite errors. The resulting significance ($p = 0.00540$) and an operational effect ratio of 1.54 mathematically validate the feature-level performance split. This statistically significant directional asymmetry indicates that the ontology-derived quality features contribute predictive information beyond the use of the filtered dataset alone.

The comparison with V1-Full provides additional supporting evidence. V2-Ontology-35770 corrected 61 errors made by V1-Full, whereas V1-Full corrected 29 errors made by V2-Ontology-35770, resulting in ($p = 0.00097$) and an effect ratio of 2.10. Across the ten size-matched random controls, the effect ratios ranged from 1.49 to 2.41, and all exact p -values were below 0.05. These results indicate that arbitrary dataset reduction does not reproduce the same paired decision behaviour. However, they do not provide a variance estimate for repeated V2 training runs.

The two control experiments answer different methodological questions. The size-matched random controls examine whether the observed results can be explained by dataset reduction alone. The direct comparison with V2-Filtered-NoOntologyFeatures holds the filtered corpus constant and changes only the ontology-derived quality features. The direct same-corpus comparison therefore provides stronger evidence for the contribution of the validation-derived feature set.

McNemar's test evaluates paired binary decisions at the validation-selected thresholds. It does not measure the complete probability ranking or performance across the full precision–recall curve. Accordingly, the significant paired difference should be interpreted together with the observed reduction in precision and Average Precision.

6.4. Limitations

Structurally, the current implementation processes each CV as a static snapshot anchored to a single point in time. Operational professional profiles, however, function as evolving histories characterized by the continuous accumulation of skills, certifications, and cross-role transitions. While the developed ontology maps the document state at the time of submission, it does not integrate the underlying temporal dynamics of career progression.

A necessary trajectory for this framework involves redefining candidate representations as a sequence of validated profile states rather than a single instance. This architectural shift enables the modelling of career trajectories and the tracking of how qualifications and responsibilities evolve across successive updates. Within this framework, temporal learning mechanisms can weigh events based on recency. Consequently, newly acquired cloud or artificial intelligence competencies, or recent shifts into advanced leadership positions, could influence the matching process differently from older professional events.

This time-dependent relevance model mirrors strategies from external domains, such as using deep LSTM architectures with temporal attention to track clinical indicators over extended periods [35]. Adapting this predictive logic from biomedicine to talent acquisition could extend the current static CV–job comparison toward the modelling of candidate career trajectories. However, this extension remains outside the scope of the present implementation and would require longitudinal data containing multiple validated CV updates for the same candidate.

A secondary constraint stems from the update latency inherent in the ESCO taxonomy. While the vocabulary offers an indispensable multilingual baseline for semantic interoperability, emerging technical sectors continuously generate new frameworks, libraries, and specialized roles that escape immediate standardization. The proposed architecture miti-

gates this latency by combining ESCO alignments with a localized domain vocabulary and isolating low-confidence concepts. However, this defensive filtering transfers operational overhead to the periodic maintenance and manual calibration of the local dictionary.

Another aspect that deserves attention is fairness. The semantic validation process may not affect all candidate profiles in the same way. CVs that use standardized terminology, recognized certifications, complete employment records, and concepts already covered by ESCO are generally easier to validate than profiles describing informal learning, career interruptions, emerging technologies, or qualifications that fall outside widely adopted taxonomies. As a result, the semantic quality indicators may be influenced not only by the quality of the extracted information but also by the level of standardization of the candidate's background.

This is particularly relevant for the training-eligibility mechanism. A profile containing uncommon terminology or concepts that cannot be mapped confidently to the ontology may receive a lower semantic-confidence score even when the candidate possesses the required knowledge and skills. The proposed system therefore reflects the current coverage of the reference vocabulary as well as the available evidence in the CV.

The datasets used in this study do not contain the demographic, socioeconomic, or protected-attribute information required for a reliable subgroup analysis. Accordingly, no claim is made that the proposed system produces equivalent error rates across candidate groups. Future work should examine false-negative rates, training-eligibility exclusions, unverified concept ratios, and classifier outcomes across relevant subgroups. Human review should remain available for profiles that are excluded or assigned low semantic confidence.

This protective firewall is structurally mandated by the system's operational domain: recruiting and talent-selection engines constitute high-risk deployments under Annex III of Regulation (EU) 2024/1689 (https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=OJ:L_202401689, accessed on 5 July 2026), enforcing strict compliance standards on automated pipelines.

Furthermore, system performance remains bound to the structural integrity of the initial ingestion layer. Although SHACL validation rules and ontological constraints isolate categorical mismatches, missing evidence, or invalid offsets, they cannot neutralize errors introduced during document parsing or named entity recognition (NER). An incorrectly extracted concept will still clear the semantic validation layer if its surrounding data properties remain internally consistent. In these scenarios, the extracted assertion persists as semantically logical while remaining factually erroneous.

Rather than serving as a guarantee of flawless parsing accuracy, the ontology functions strictly to manage semantic uncertainty. Advancing this framework requires embedding human-in-the-loop auditing directly into the validation stream, alongside probabilistic calibration for the NER pipeline. Introducing active learning loops targeted at ambiguous boundary cases could reduce the propagation of upstream extraction errors and support the periodic refinement of both the local vocabulary and the validation rules.

The scope of the external evaluation remains limited. Although the benchmark contains 2520 CV–job pairs, these pairs correspond to only 63 unique candidates and 125 job descriptions from an IT-focused subset. The reported results should therefore not be generalized automatically to other occupational sectors, languages, CV formats, or candidate populations. Further evaluation on additional independently labelled benchmarks is required.

The same-corpus ablation also uses the fixed experimental seed adopted throughout the study. This provides a controlled comparison between V2-Filtered-NoOntologyFeatures and V2-Ontology-35770, because both models use identical pairs, labels, splits, and modelling procedures. However, it does not quantify variability caused by repeated model

initialization or stochastic subsampling. The significant McNemar result applies to the evaluated runs, while future experiments should repeat both configurations across multiple training seeds.

Recruiter-guided score reweighting and adaptive assessment generation are retained only as future application-level extensions. These mechanisms were not used in the model-training process or in the benchmark experiments reported in this study. Their effect should be evaluated separately through controlled ranking experiments, recruiter studies, and candidate-assessment validation.

7. Conclusions

The present paper shows that ensuring the semantic consistency of data in resumes is critical, and that increasing the volume of extracted information alone is not sufficient when developing candidate matching systems. Even though Natural Language Processing (NLP) techniques facilitate the large-scale extraction of candidate profiles, the absence of standardization and the presence of semantic inconsistencies can have a substantial impact on the reliability of the classification process.

The proposed ontological approach addresses this issue by structuring the extracted data using RDF representations aligned with ESCO and validating them through SHACL constraints. In contrast to conventional filtering strategies, the system does not eliminate all non-compliant data. Instead, it introduces a training eligibility mechanism that allows for the retention of semantically incomplete but recoverable profiles. This facilitates the model's capacity to acquire knowledge from both valid and near-valid cases, thereby reflecting more realistic recruitment scenarios.

The incorporation of signals derived from the ontology, including validation status, semantic inconsistencies and unverified concepts, into the classification stage of the model results in the acquisition of an additional layer of semantic context. The combination of these signals with the class-level matching features used by XGBoost supports a more interpretable classification process, because the final score can be related not only to CV-job compatibility, but also to the semantic reliability of the extracted evidence.

Comparing two filtered-corpus configurations allows us to examine the contribution of quality indicators derived from the ontology without modifying the training data. Both V2-Filtered-NoOntologyFeatures and V2-Ontology-35770 were trained on the same 35,770 CV-job pairs, but the model that includes semantic quality indicators improves performance on the external benchmark, as accuracy and recall reach 0.809 and 0.865, respectively, compared to the 0.794 and 0.760 values obtained by the configuration without the semantic indicators. This performance difference is based on an explicit operational trade-off: precision drops to 0.720 from 0.739, and Average Precision reaches 0.844 compared to the baseline value of 0.851.

Because the benchmark contains only 63 unique candidates from an IT-focused subset and the direct ablation uses a fixed training seed, the findings should remain scoped to the evaluated setting until repeated-seed experiments and additional independent benchmarks become available.

The system has been demonstrated to exhibit the capacity to integrate structured domain knowledge into the data processing workflow, thereby engendering more traceable and interpretable matching results. At the same time, the revised double-disjoint evaluation indicates that previous internal results were likely too optimistic; therefore, the current findings should be interpreted as evidence of improved semantic traceability and selected-threshold robustness, not as proof that overfitting has been eliminated. This, in turn, supports the development of more effective recruitment decision-support systems.

In future, the proposed framework could be extended in several directions. One potential enhancement to consider is the integration of advanced semantic reasoning capabilities within the ontology. This would facilitate the automated identification of implicit relationships between competencies, roles and learning pathways. Another promising development is the integration of dynamic feedback mechanisms derived from recruiter interactions, allowing the system to adapt the importance weights and validation strategies over time. Furthermore, the uncertainty handling mechanism based on unverified concepts could be refined through adaptive functions or probabilistic reasoning models instead of fixed penalty factors. Future work should also evaluate the framework on additional independent benchmarks and investigate temporal extensions in which multiple CV updates are modelled as candidate career trajectories rather than as static snapshots.

Beyond recruitment, the broader data-structuring and uncertainty-management principles examined in this study may also be investigated in other resource-allocation domains. While the current study utilizes machine learning and ontologies to optimize human resource allocation, future research could adapt these data-structuring techniques to physical resource optimization, such as land-use constraints in clean-energy expansion. For instance, similar predictive models could be employed to assess the trade-offs between clean-energy deployment and spatial limitations, as explored in grid capacity energy modelling systems [36].

Future iterations of the CV classification model could benefit from exploring alternative, data-driven probability distributions for handling candidate skill uncertainty, moving beyond standard heuristic thresholds. Applying methodologies like Kernel Density Estimation (KDE) and stochastic optimization, which have proven successful in managing economic risk in smart energy hubs, could further refine the model's predictive accuracy [37].

Finally, the framework's ability to handle ambiguous data could be further strengthened by explicitly modelling both exogenous and endogenous uncertainties within the recruitment lifecycle. Incorporating stochastic optimization approaches utilized in smart grid technology planning could allow recruiters to calculate the strategic option value of specific hires under various future business scenarios [38].

Within the recruitment domain, future work should evaluate the framework on additional independently labelled benchmarks, repeat the same-corpus ablation across multiple training seeds, and investigate subgroup fairness, including the effects of incomplete profiles, non-standard career paths, unequal access to standardized credentials, and limited ESCO coverage. Further research should also evaluate recruiter-guided ranking adjustments and adaptive assessment generation through dedicated experiments rather than treating them as validated components of the current pipeline.

These directions could support the development of recruitment decision-support systems that are more adaptive, explainable, semantically aware, and sensitive to the limitations of automated candidate evaluation.

Author Contributions: Conceptualization, V.-V.A., T.B., S.R. and C.N.; Methodology, V.-V.A., T.B., S.R. and C.N.; Software, V.-V.A.; Validation, T.B. and S.R.; Formal analysis, C.N.; Investigation, S.R. and C.N.; Resources, V.-V.A. and C.N.; Data curation, T.B. and S.R.; Writing—original draft preparation, V.-V.A.; Writing—review and editing, T.B., S.R. and C.N.; visualization, V.-V.A., T.B. and S.R.; supervision, T.B.; project administration, V.-V.A. and T.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Panchendrarajan, R.; Zubiaga, A. Synergizing machine learning & symbolic methods: A survey on hybrid approaches to natural language processing. *Expert Syst. Appl.* **2024**, *251*, 124097. [\[CrossRef\]](#)
- de Groot, M.; Schutte, J.; Graus, D. Job Posting-Enriched Knowledge Graph for Skills-Based Matching. In *Proceedings of the Workshop on Recommender Systems for Human Resources, RecSys in HR 2021, Amsterdam, The Netherlands, 1 October 2021*; CEUR Workshop Proceedings; ACM: New York, NY, USA, 2021; Volume 2967. Available online: <https://ceur-ws.org/Vol-2967/> (accessed on 15 May 2026).
- Tuan, A.C.; Dang, M.T.; Do, H.N.; Solanki, V.K.; Torres, J.; Gonzalez Crespo, R.; Nguyen, T.N.A. Ontology and its applications in skills matching in job recruitment. *Appl. Ontol. Interdiscip. J. Ontol. Anal. Concept. Model.* **2024**, *19*, 287–306. [\[CrossRef\]](#)
- Wasi, A.T. HRGraph: Leveraging LLMs for HR Data Knowledge Graphs with Information Propagation-based Job Recommendation. In *Proceedings of the 1st Workshop on Knowledge Graphs and Large Language Models (KaLLM 2024)*; Conference Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 56–62. [\[CrossRef\]](#)
- Tang, F.; Zhu, R.; Yao, F.; Wang, J.; Luo, L.; Li, B. Explainable person–job recommendations: Challenges, approaches, and comparative analysis. *Front. Artif. Intell.* **2025**, *8*, 1660548. [\[CrossRef\]](#) [\[PubMed\]](#)
- Phan, T.T.; Pham, V.Q.; Nguyen, H.D.; Huynh, A.T.; Tran, D.A.; Pham, V.T. Ontology-Based Resume Searching System for Job Applicants in Information Technology. In *Advances and Trends in Artificial Intelligence. Artificial Intelligence Practices. IEA/AIE 2021*; Fujita, H., Selamat, A., Lin, J.C.W., Ali, M., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2021; Volume 12798. [\[CrossRef\]](#)
- Habous, A.; Nfaoui, E.H. A fuzzy logic and ontology-based approach for improving the CV and job offer matching in recruitment process. *Int. J. Metadata Semant. Ontol.* **2021**, *15*, 104–120. [\[CrossRef\]](#)
- Fettach, Y.; Bahaj, A.; Ghogho, M. JobEdKG: An uncertain knowledge graph-based approach for recommending online courses and predicting in-demand skills based on career choices. *Eng. Appl. Artif. Intell.* **2024**, *131*, 107779. [\[CrossRef\]](#)
- Giabelli, A.S.; Malandri, L.; Mercorio, F.; Mezzanzanica, M.; Seveso, A. Skills2Job: A recommender system that encodes job offer embeddings on graph databases. *Appl. Soft Comput.* **2021**, *2021*, 107049. [\[CrossRef\]](#)
- Rosenberger, J.; Wolfrum, L.; Weinzierl, S.; Kraus, M.; Zschech, P. CareerBERT: Matching Resumes to ESCO Jobs in a Shared Embedding Space for Generic Job Recommendations. *arXiv* **2025**, arXiv:2503.02056.
- Clavie, B.; Soulie, G. Large Language Models as Batteries-Included Zero-Shot ESCO Skills Matchers. *arXiv* **2023**, arXiv:2307.03539.
- Zhang, M.; van der Goot, R.; Plank, B. ESCOXML-R: Multilingual Taxonomy-driven Pre-training for the Job Market Domain. *arXiv* **2023**, arXiv:2305.12092.
- Rabbani, K.; Lissandrini, M.; Hose, K. SHACL and ShEx in the Wild: A Community Survey on Validating Shapes Generation and Adoption. In *WWW'22: Companion Proceedings of the Web Conference 2022*; ACM: New York, NY, USA, 2022. [\[CrossRef\]](#)
- Cimmino, A.; Fernández-Izquierdo, A.; García-Castro, R. Astrea: Automatic Generation of SHACL Shapes from Ontologies. In *The Semantic Web (ESWC 2020)*; LNCS; Springer: Berlin/Heidelberg, Germany, 2020. [\[CrossRef\]](#)
- Pareti, P.; Konstantinidis, G.; Mogavero, F. Satisfiability and Containment of Recursive SHACL. *J. Web Semant.* **2022**, *74*, 100721. [\[CrossRef\]](#)
- Knublauch, H.; Kontokostas, D. Shapes Constraint Language (SHACL). *W3C Recommendation*. 20 July 2017. Available online: <https://www.w3.org/TR/shacl/> (accessed on 10 May 2026).
- Pareti, P.; Konstantinidis, G. *A Review of SHACL: From Data Validation to Schema Reasoning for RDF Graphs*; Lecture Notes in Computer Science; Springer International Publishing: Berlin/Heidelberg, Germany, 2022; pp. 115–144. [\[CrossRef\]](#)
- Qin, C.; Zhu, H.; Xu, T.; Zhu, C.; Jiang, L.; Chen, E.; Xiong, H. Enhancing Person-Job Fit for Talent Recruitment: An Ability-aware Neural Network Approach. *arXiv* **2018**, arXiv:1812.08947.
- Wang, Z.; Wei, W.; Xu, C.; Xu, J.; Mao, X.-L. Person-job fit estimation from candidate profile and related recruitment history with co-attention neural networks. *arXiv* **2022**, arXiv:2206.09116.
- Yu, X.; Xu, R.; Xue, C.; Zhang, J.; Ma, X.; Yu, Z. ConFit v2: Improving Resume-Job Matching Using Hypothetical Resume Embedding and Runner-Up Hard-Negative Mining; Findings of the Association for Computational Linguistics: ACL 2025; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 12775–12790. [\[CrossRef\]](#)
- Chen, H.; Du, L.; Lu, Y.; Fu, Q.; Chen, X.; Han, S.; Kang, Y.; Lu, G.; Li, Z. Professional Network Matters: Connections Empower Person-Job Fit. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*; ACM: New York, NY, USA, 2024; pp. 96–105. [\[CrossRef\]](#)
- Lo, F.P.-W.; Qiu, J.; Wang, Z.; Yu, H.; Chen, Y.; Zhang, G.; Lo, B. AI Hiring with LLMs: A Context-Aware and Explainable Multi-Agent Framework for Resume Screening. *arXiv* **2025**, arXiv:2504.02870.
- Lundberg, S.M.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, 4–9 December 2017.

24. Vyaas, M.; Chakrabroty, A.; Gupta, V. JobMatchAI: An Intelligent Job Matching Platform Using Knowledge Graphs, Semantic Search and Explainable AI. *arXiv* **2026**, arXiv:2603.14558.
25. Phenom. X+ Ontologies—Unified Talent Data Intelligence. 2026. Available online: <https://www.phenom.com/x-plus-ontologies-ai> (accessed on 12 May 2026).
26. Phenom. Aptitude Research. State of Hiring Automation: 2026 Benchmark Report. 2026. Available online: <https://www.phenom.com/state-ai-automation-report-2026> (accessed on 12 May 2026).
27. Ke, J.; Zacouris, Z.; Acosta, M. Efficient Validation of SHACL Shapes with Reasoning. *Proc. VLDB Endow.* **2024**, *17*, 3589–3601. [[CrossRef](#)]
28. Rabbani, K.; Lissandrini, M.; Hose, K. Extraction of Validating Shapes from Very Large Knowledge Graphs. *Proc. VLDB Endow.* **2023**, *16*, 1023–1032. [[CrossRef](#)]
29. Waseem, M.; Liang, P. Microservices Architecture in DevOps. In *Proceedings of the 2017 24th Asia-Pacific Software Engineering Conference Workshops (APSECW), Nanjing, China, 4–8 December 2017*; IEEE: Piscataway, NJ, USA, 2017; pp. 13–14. [[CrossRef](#)]
30. Chen, L. Microservices: Architecting for Continuous Delivery and DevOps. In *Proceedings of the 2018 IEEE International Conference on Software Architecture (ICSA), Seattle, WA, USA, 30 April–4 May 2018*; IEEE: Piscataway, NJ, USA, 2018; pp. 39–397. [[CrossRef](#)]
31. RabbitMQ. Consumer Acknowledgements and Publisher Confirms. Version 4.3, RabbitMQ Documentation. 2026. Available online: <https://www.rabbitmq.com/docs/confirms> (accessed on 11 May 2026).
32. Rahmatulloh, A.; Nugraha, F.; Gunawan, R.; Darmawan, I. Event-Driven Architecture to Improve Performance and Scalability in Microservices-Based Systems. In *Proceedings of the 2022 International Conference Advancement in Data Science, E-Learning and Information Systems (ICADEIS), Bandung, Indonesia, 23–24 November 2022*; IEEE Xplore: Piscataway, NJ, USA, 2022; pp. 1–6. [[CrossRef](#)]
33. PostgreSQL Global Development Group. JSON Types. In PostgreSQL Documentation, Version 18. Available online: <https://www.postgresql.org/docs/current/datatype-json.html> (accessed on 17 May 2026).
34. Walls, C. *Spring Boot in Action*; Manning Publications Co.: Shelter Island, NY, USA, 2016.
35. Cai, J.; Li, Y.; Liu, B.; Wu, Z.; Zhu, S.; Chen, Q.; Lei, Q.; Hou, H.; Guo, Z.; Jiang, H.; et al. Developing Deep LSTMs with Later Temporal Attention for Predicting COVID-19 Severity, Clinical Outcome, and Antibody Level by Screening Serological Indicators over Time. *IEEE J. Biomed. Health Inform.* **2024**, *28*, 4204–4215. [[CrossRef](#)] [[PubMed](#)]
36. Koutsandreas, D.; Ardehali, A.; Giannelos, S.; Pudjianto, D. Quantifying the Trade-Offs Between Clean-Energy Expansion and Land Requirements: Evidence from Greece. *Energies* **2026**, *19*, 2261. [[CrossRef](#)]
37. Giannelos, S.; Pudjianto, D.; Strbac, G. Smart home economic operation under uncertainty: Comparing Monte Carlo and stochastic optimization using Gaussian and KDE-based data. *Oper. Res. Perspect.* **2025**, *15*, 100348. [[CrossRef](#)]
38. Giannelos, S. Option Valuation of Smart Grid Technology Projects Under Endogenous and Exogenous Uncertainty. Doctoral Dissertation, Imperial College London, London, UK, 2016.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.