Methodology for Converting and Publish Tabular Data into SKOS Resources via Python Notebooks

Michele Mallia
CNR-ILC
Pisa, Italy
michele.mallia@ilc.cnr.it

Fahad Khan
CNR-ILC
Pisa, Italy
fahad.khan@ilc.cnr.it

Silvia Calvi
Osservatorio di Terminologie
e Politiche Linguistiche
Università Cattolica
del Sacro Cuore, Italy
silvia.calvi1@unicatt.it

Klara Dankova
Osservatorio di Terminologie
e Politiche Linguistiche
Università Cattolica
del Sacro Cuore, Italy
klara.dankova@unicatt.it

Abstract

This paper presents a methodology for creating and converting tabular data into SKOS linguistic resources using Python notebooks. Designed to support users with limited technical skills, the approach offers a structured and reproducible process through interactive notebooks. The methodology covers metadata preparation, data scraping from repositories, information mapping, and data normalization for standardized vocabulary publication. Various specialized multilingual vocabularies, including those related to textile description and smart city terminology, were analyzed to evaluate the approach. Guidelines were also developed to optimize LOD environment deployment, including resource uploading and web application configuration. The methodology supports linguistic data management as Linked Open Data, offering a platform for hosting SKOS resources and training users to create structured data efficiently. The system was tested through external user engagement, demonstrating its scientific relevance and practical utility.

1 Introduction

The increasing adoption of Linked Open Data (LOD) practices in linguistic resource management highlights the need for efficient and accessible methods to convert tabular data into RDF[1]. While progress has been made in promoting Linked Data principles, transforming non-standardised formats such as CSV or TSV into SKOS[2] remains a challenge. Existing tools (Fiorelli & Stellato, 2021) often require advanced technical skills and are not always user-friendly. OpenRefine[3] stands out as a powerful tool for data transformation, but lacks a training-oriented approach.

To address this gap, we propose a methodology based on Python Jupyter notebooks[4], designed to guide non-expert users step by step through the process of converting tabular data into SKOS. This approach combines user accessibility with data interoperability and quality, serving both as a practical tool and a training resource. Unlike general-purpose tools or language models like ChatGPT, our notebooks offer structured, domain-specific, and reproducible workflows aligned with LOD best practices.

The methodology has been tested in a real-world use case as part of a pilot initiative (described in Section 2.1), where external users applied the notebooks in a research context. This evaluation confirmed their usability and potential for broader adoption within linguistic data platforms. In addition, this workflow was used to represent and disseminate terminological data from the multilingual lexicons of the Pan-Latin Terminology Network – REALITER[5] in compliance with FAIR principles, thanks to the col-

[1] https://www.w3.org/RDF/
[2] https://www.w3.org/2004/02/skos/
[3] https://openrefine.org/
[4] https://github.com/cnr-ilc/ilc4clarin-jupyter-skos-notebook
[5] https://www.realiter.net/

laboration between the CNR-ILC's B center of ILC4CLARIN - CLARIN-IT [6] (the Italian national node of CLARIN ERIC [7], the European infrastructure for language resources and technologies) and OTPL (Osservatorio di Terminologie e Politiche Linguistiche, Università Cattolica del Sacro Cuore – member of CLARIN-IT). More specifically, the REALITER – OTPL collection in the CLARIN-IT repository has been created and the lexicons have been published on the ILC4CLARIN - Skosmos Service platform.

The remainder of this paper is structured as follows: Section 2provides an overview of related work and contextual background. Section 3 details the methodology, including data acquisition, transformation, and deployment. Section 4presents case studies of REALITER lexicon to illustrate the practical application of this approach; in this section we discuss also the results and challenges encountered, followed by conclusions and suggestions for future work in Section 5.

2 Background and context

The landscape of linguistic data management is increasingly shaped by the principles of Linked Open Data (Berners-Lee, 2006), which aim to enhance the accessibility, interoperability, and usability of structured data across digital platforms. The integration of linguistic data into LOD frameworks is particularly significant, as it fosters the development of interconnected language resources, contributing to a more dynamic and semantically enriched web of data (Khan et al., 2022).In the following article we will focus on resources which are based on the SKOS vocabulary[8], which are part of the Linguistic Linked Open Data Cloud[9] under the category of 'Terminologies, Thesauri and Knowledge Bases'.

2.1 H2IOSC LLOD Pilot

The project described in this paper is part of a broader pilot initiative within the H2IOSC project[10] (Humanities and Heritage Italian Open Science Cloud), which aims to promote open and FAIR(Lamprecht et al., 2020) access to services and data in the domain of humanities and cultural heritage. As part of this initiative, the LLOD pilot[11] (Linguistic Linked Open Data pilot) focuses on advancing LLOD technologies for managing and disseminating linguistic resources in interoperable and semantically rich formats. The LLOD pilot is developing a comprehensive infrastructure to support the entire lifecycle of linguistic data as Linked Data. This includes components for the transformation of tabular or XML-based resources into RDF, tools for browsing and editing language data, and services for publishing and querying linked data via SPARQL endpoints. The platform also integrates with Skosmos for vocabulary browsing, and features a quality assessment dashboard to ensure data consistency, adherence to LOD principles, and compliance with vocabularies such as OntoLex-Lemon[12] and SKOS.

To ensure long-term accessibility and sustainability, resources are ingested into a central CLARIN repository[13], which supports version control, metadata management, and integration into the broader European research infrastructure. Moreover, the pilot places strong emphasis on training and community support, providing tutorials, interactive Jupyter Notebooks for data conversion, and dedicated assistance to facilitate adoption by researchers, educators, and institutions. Through this approach, the LLOD pilot within H2IOSC contributes to the creation of a national infrastructure for linguistic Linked Data that is open, reusable, and aligned with European Open Science practices.

[6] https://www.clarin-it.it/it

[7] https://www.clarin.eu/

[8] https://www.w3.org/TR/swbp-skos-core-spec/

[9] https://linguistic-lod.org/

[10] H2IOSC Project – Humanities and cultural Heritage Italian Open Science Cloud funded by the European Union NextGenerationEU – National Recovery and Resilience Plan (NRRP) – Mission 4 "Education and Research" Component 2 "From research to business" Investment 3.1 "Fund for the realization of an integrated system of research and innovation infrastructures" Action 3.1.1 "Creation of new research infrastructures strengthening of existing ones and their networking for Scientific Excellence under Horizon Europe" – Project code IR0000029 – CUP B63C22000730005. Implementing Entity CNR. https://www.h2iosc.cnr.it/

[11] https://pllod.clarin-it.it/

[12] https://www.w3.org/2019/09/lexicog/

[13] https://dspace-clarin-it.ilc.cnr.it/repository/xmlui/

3 Methodology

In this section, we describe a comprehensive pipeline for creating SKOS resources from tabular data, focusing on the transformation and publication processes. The workflow is designed as a modular and reproducible sequence of steps that guides users from the initial acquisition of metadata to the final deployment of the generated RDF vocabulary. Particular attention is given to ensuring that the conversion from tabular formats to SKOS remains transparent, standards-compliant, and suitable for integration into LLOD-oriented infrastructures. The following subsections outline the main components of this pipeline and illustrate how the methodology supports both technical processing and user-oriented training activities.

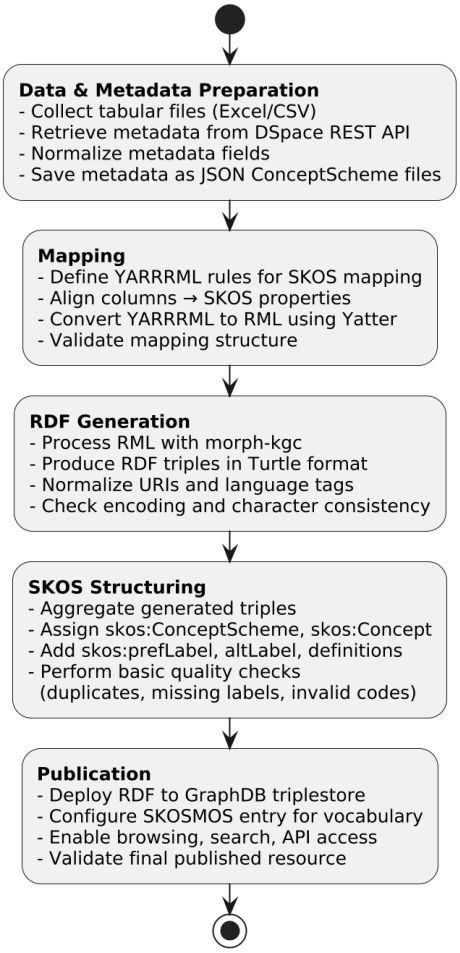


Figure 1: Overview of the SKOS conversion workflow, from data preparation to publication.

3.1 Overview of the LLOD Pipeline

This methodology outlines a streamlined approach for creating SKOS-formatted linguistic resources. The workflow begins with the collection and normalisation of metadata retrieved from institutional repositories, ensuring that each vocabulary is associated with consistent descriptive information before conversion. Once metadata has been harmonised, a manual alignment step is carried out to associate the structured data files (e.g. CSV, TSV, XLSX) with their corresponding metadata record. This non-programmatic association allows for precise matching between the conceptual scope of the resource and the structure of the tabular content, enabling the construction of an initial SKOS `ConceptScheme`.

Following this alignment, the cleaned structured data is transformed into RDF triples through a mapping-based approach. The pipeline employs human-readable YARRRML[14] specifications to define

[14]https://rml.io/yarrrml/

how tabular elements should be projected onto SKOS properties, and these specifications are then converted into executable RML mappings. The resulting RML files are processed using the `morph-kgc` engine[15], which generates the final Turtle serialisation while ensuring consistency in language tagging, encoding, and URI formation.

Once the RDF has been produced, the resulting dataset is prepared for deployment on standard knowledge-graph infrastructures. Depending on the needs of the data providers, the output can be loaded into a triplestore—such as GraphDB—and exposed through a SKOS-compliant interface such as SKOSMOS. This final publication stage enables efficient exploration, search, and reuse of the newly created lexical resources, ensuring their integration into broader LLOD ecosystems.

3.2 Data Acquisition and Preparation

This phase of the workflow focuses on gathering, normalising, and structuring the information required to support the creation of SKOS-formatted vocabularies. Data acquisition is carried out in two distinct but complementary steps: (i) the retrieval and preparation of metadata describing the vocabulary resource, and (ii) the ingestion and cleaning of the tabular files containing the actual lexical content. Together, these operations ensure that both descriptive and structural information are consistently represented before the RDF transformation process begins.

3.2.1 Metadata

The first component of the acquisition process concerns the retrieval and preparation of metadata associated with each vocabulary. In our implementation, metadata are sourced from the DSpace repository of ILC4CLARIN through its REST API. The pipeline identifies the target collection (e.g. the REALITER terminology network), retrieves its items, and accesses the corresponding item-level metadata records.

These records include core descriptive elements such as authorship, licensing information, language, date of issuance, subject classification, and persistent identifiers (e.g. Handles). Metadata keys are standardised, multi-valued fields are grouped, and the records are reshaped into structured JSON descriptors. Particular attention is given to fields such as `dc.language.iso`, which are converted into explicit language objects for later use in the SKOS `ConceptScheme`.

The resulting JSON files act as authoritative descriptors for each vocabulary, ensuring that the subsequent RDF generation is aligned with the metadata deposited in the institutional repository and providing a solid foundation for the remaining phases of the conversion pipeline.

3.2.2 Source Files

The second component of the preparation workflow concerns the ingestion and normalisation of the structured files containing the lexical content of each vocabulary. Unlike metadata—retrieved automatically from the institutional repository—these tabular files are provided separately by data producers and manually uploaded into the Jupyter Notebook environment. Typically distributed in Excel or CSV format, they must be aligned with the previously extracted metadata to ensure a consistent and reproducible transformation into SKOS.

To guarantee a correct association between each tabular dataset and its corresponding metadata record, source files are manually renamed according to the identifier of the `ConceptScheme`. This ensures that metadata JSON files and structured data share a common base name, allowing the notebook to automatically link them during processing. Once harmonised, the structured data undergoes a series of automated normalisation procedures implemented within the notebook.

The notebook includes dedicated routines for cleaning column names, handling language variants, and standardising the dataset structure. Column headers are processed using the `clean_column_name` function, which unifies naming conventions, resolves regional language notations (e.g. `es-MX`, `pt-BR`, `fr-CA`), and removes problematic characters. Each row is then analysed to extract lexical elements such as preferred and alternative labels, definitions, and notes, while concept identifiers are sanitised through `clean_concept_name` to ensure URI-compatible strings.

[15] https://github.com/morph-kgc/morph-kgc

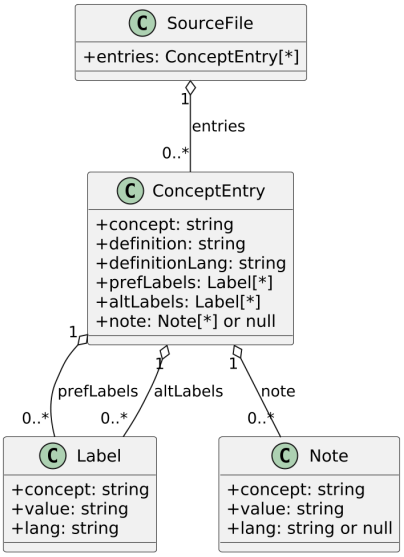


Figure 2: Schema of the JSON used to represent concepts information extracted from the source files.

The processed contents are restructured into a uniform JSON representation including concept identifiers, multilingual labels, definitions, and notes. These intermediate JSON files provide a normalised, machine-readable representation of the vocabulary, enabling consistent handling of heterogeneous datasets and preparing them for the semantic lifting procedures described in the following sections.

3.3 Preparation of the YARRRML Mapping Files

Once the metadata descriptors and the normalised source files have been generated, the next phase of the workflow consists in preparing the YARRRML mapping files that define how input data is translated into RDF triples. In this pipeline, YARRRML acts as an intermediate, human-readable layer that abstracts the complexity of RML while retaining fine-grained control over the mapping logic. Each mapping file corresponds to a single vocabulary and specifies the rules required to produce a coherent SKOS representation from its metadata and lexical content.

The construction of each mapping file starts by pairing a ConceptScheme descriptor with its corresponding structured data file. This pairing relies on filename alignment: once metadata and source files share a common identifier derived from the repository's persistent URI, they can be reliably processed together. From the metadata, key parameters are extracted, most notably the `dc.identifier.uri`, which is used to derive a custom namespace for the vocabulary. This namespace serves as the base for all generated SKOS concept URIs, ensuring stability and predictability.

The mapping file is organised into two main sections. The first defines the ConceptScheme itself by mapping metadata properties—such as title, description, identifier, creation date, authorship, and languages—to Dublin Core and SKOS predicates. This section establishes the structural container for the vocabulary and ensures that provenance, licensing, and linguistic information are correctly represented.

The second section specifies the transformation of the structured data into SKOS concepts. Using JSONPath expressions over the normalised JSON produced in the preprocessing phase, the mapping defines how concepts, labels, definitions, and notes are materialised as RDF triples. For each lexical entry, a URI is generated under the vocabulary namespace and enriched with the appropriate SKOS properties. Language-sensitive fields are handled explicitly, with language tags assigned to labels and notes and inherited by definitions from their source columns.

To ensure interoperability, the mapping includes standard prefixes (e.g. `skos`, `dc`, `dct`, `xsd`) together with a dynamically generated prefix representing the vocabulary namespace. The resulting YARRRML

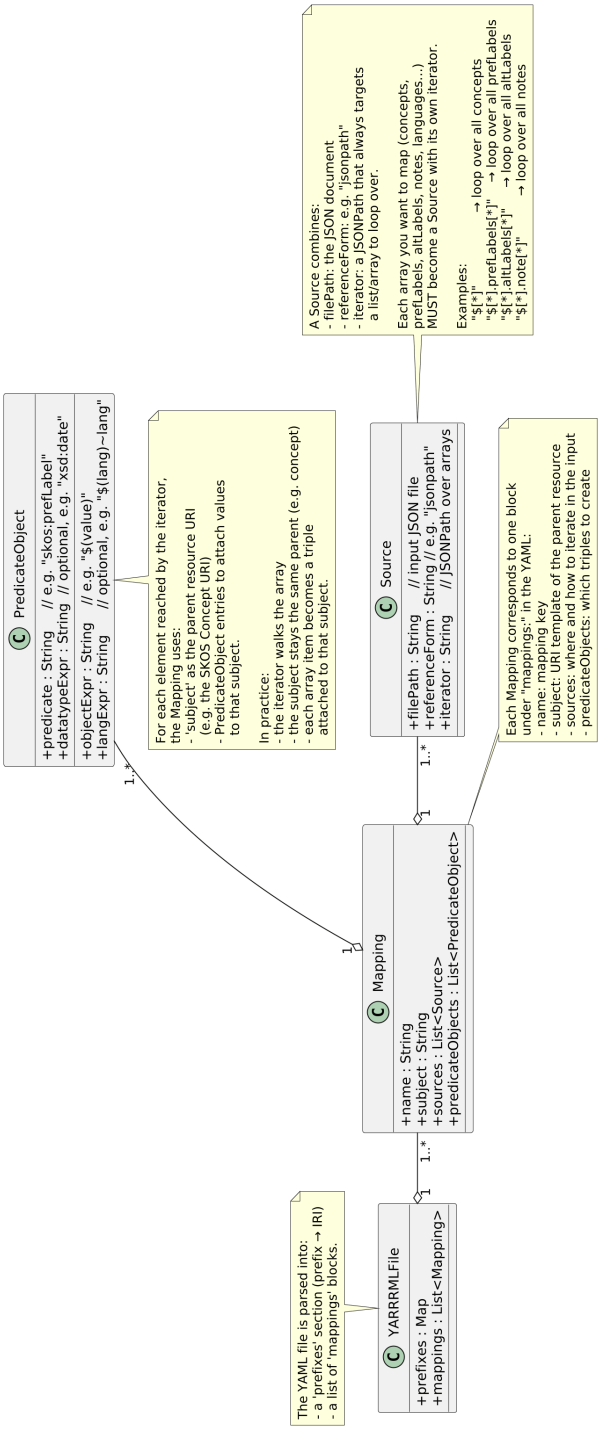


Figure 3: YARRRML Mapping Logic for Concepts.

files, saved as individual YAML documents in the `mapping_files` directory, provide a declarative and reproducible specification of the transformation logic. These files are subsequently converted into RML and processed with `morph-kgc` to generate the final RDF representation described in the next section.

3.4 Creation of RML Files

After defining the YARRRML mappings for each vocabulary, the next step of the pipeline consists in translating these human-readable specifications into RML (RDF Mapping Language) files. This phase introduces an intermediate representation that can be directly consumed by RDF generation engines, while preserving the structure and semantics defined in the YARRRML layer.

The conversion is performed using the Yatter library, which takes YARRRML YAML files as input and produces equivalent RML mappings serialised in Turtle syntax. The notebook iterates over all mapping files stored in the `mapping_files` directory, parses each `.yaml` document, and passes it to Yatter for translation. For each vocabulary, a corresponding RML file is generated, mirroring the original mapping rules in terms of sources, iterators, subject templates, and predicate–object definitions.

During this step, the pipeline also normalises the RML namespace by replacing the legacy namespace (`http://semweb.mmlab.be/ns/rml#`) with its updated counterpart (`http://w3id.org/rml/`), ensuring compliance with current best practices and facilitating integration with contemporary tools. The resulting RML documents are stored in the `rml_files` directory, preserving the base filename of the original YARRRML file and adopting the `.rml` extension.

By separating mapping authoring (YARRRML) from execution (RML), this phase increases transparency and modularity. The resulting RML files provide a stable and reusable specification that can be repeatedly applied to the prepared JSON sources to generate RDF triples in a controlled and reproducible manner, as described in the subsequent phase of the pipeline.

3.5 Creation of RDF Files

The final stage of the transformation pipeline consists in generating RDF data from the RML mapping files created in the previous step. To perform this operation we rely on the `morph-kgc` library, a Python framework capable of executing R2RML, RML, and RML-star mappings and materialising the corresponding RDF triples. This step produces the final, serialised RDF representation of each vocabulary, ready to be published and ingested into standard Linked Data platforms.

3.5.1 RDF Generation Workflow

The creation of RDF files follows a unified workflow that combines configuration, execution, and serialisation into a single processing phase. First, a configuration string is generated for each RML file, specifying the input mapping, the desired output format, and the destination directory for storing the resulting RDF data. This dynamic configuration replaces the need for a static `config.ini` file and ensures consistent processing across all vocabularies.

Once the configuration is prepared, the notebook iterates over the RML files contained in the `rml_files` directory. For each mapping, the `morph-kgc` `materialize()` function is invoked to execute the RML rules. During this process, iterators defined in the mapping files identify the relevant elements in the structured JSON data, subject templates are expanded to generate IRIs for SKOS concepts, and predicate–object patterns produce the associated labels, definitions, and notes.

After the triples are materialised, the resulting `rdflib` graph is enriched with a predefined set of prefixes (e.g. `dc`, `dct`, `skos`) and a dynamic prefix derived from the mapping filename. This dynamic namespace corresponds to the final vocabulary URI and ensures that all IRIs in the output adhere to the expected namespace pattern. Finally, the enriched graph is serialised in Turtle format and saved to the `rdf_files` directory, yielding the complete and FAIR-compliant RDF dataset for each vocabulary.

3.5.2 Aggregating Concepts into the ConceptScheme

Once the RDF files have been generated from the RML mappings, each Turtle file contains two distinct types of resources: the `skos:ConceptScheme` describing the vocabulary as a whole, and the individual `skos:Concept` entries representing the terms defined within it. In their initial form, however, these resources remain unconnected. Establishing explicit links between the `ConceptScheme` and its concepts is therefore a necessary post-processing step to ensure the semantic completeness and navigability of the resulting SKOS resource.

This aggregation step relies on two complementary SKOS properties:

- **skos:hasTopConcept** — associates a skos:ConceptScheme with each of its top-level concepts.
- **skos:topConceptOf** — establishes the inverse relationship, linking each skos:Concept back to its parent ConceptScheme.

The procedure operates over the set of Turtle files generated in the previous step. For each file, the script identifies the unique skos:ConceptScheme resource it contains, extracts all the instances of skos:Concept, and systematically adds the bidirectional relationships skos:hasTopConcept and skos:topConceptOf. The enriched RDF graph is then serialized back into Turtle format, replacing the original file. This ensures that all vocabularies produced by the pipeline expose the hierarchical structure required for correct visualisation and browsing in SKOS-oriented platforms such as SKOSMOS.

3.5.3 Normalising Character Encoding in Concept URIs

A final refinement step is applied to ensure that all URIs contained in the generated RDF files are human-readable and free from percent-encoded characters. During previous phases, concept identifiers originating from tabular sources may contain accented or non-ASCII characters, which can become percent-encoded in the form of sequences such as %C3%A9 for é. While these encodings are technically valid, they reduce readability and hinder the use of the resulting SKOS resources in user-facing environments.

To address this issue, each Turtle file produced in the RDF generation stage is scanned for URIs containing percent-encoded segments. The normalisation process proceeds as follows:

- The file is parsed into an rdflib graph, preserving all existing namespace declarations.
- Each triple is inspected, and both subject and object nodes are checked for the presence of percent-encoded characters.
- When such patterns are detected, the URI is decoded into its UTF-8 representation using standard URL-decoding rules.
- The triple is then reconstructed using the cleaned URI, and added to a new RDF graph that retains the original prefixes.

Once all triples have been processed, the updated graph is serialised back into Turtle format and written to disk, replacing the original file. This ensures that all concept URIs are expressed in a normalised and readable form, improving interoperability and guaranteeing that multilingual labels encoded within identifiers (e.g. concept names derived directly from tabular structures) are faithfully represented in the final SKOS dataset.

3.6 Configuring SKOS Vocabularies in SKOSMOS

Once the SKOS vocabularies have been generated as RDF and loaded into the GraphDB repository, the final step consists in exposing them through the SKOSMOS web interface. This is done by adding one configuration block per vocabulary in the SKOSMOS configuration file (config.ttl or its deployment-specific variant). Each block defines a skosmos:Vocabulary resource associated with the corresponding named graph in GraphDB, allowing SKOSMOS to provide browsing, search functionalities, and API access for that specific lexicon.

3.6.1 Vocabulary Declarations

SKOSMOS requires each lexicon to be declared as an instance of both skosmos:Vocabulary and void:Dataset. In the Pan-Latin pipeline, each vocabulary receives an identifier derived from the internal handle-based name. For example, a vocabulary identified internally as 20_500_11752_OPEN_1014 is represented in the configuration as follows:

```
:pl_20_500_11752_OPEN_1014 a skosmos:Vocabulary, void:Dataset ;
    dc:title "Pan-Latin Office Supplies Vocabulary"@en ;
    skosmos:shortName "Pan-Latin Office Supplies Vocabulary" ;
    dc:subject panlatin:LexiconCat ;
    void:uriSpace
        "https://vocabs.ilc4clarin.ilc.cnr.it/vocabularies/pl_20_500_11752_OPEN_1014/" ;
    skosmos:language "it", "en", "fr", "es-AR", "pt-BR" ;
    skosmos:defaultLanguage "it" ;
    skosmos:showTopConcepts "true" ;
    void:sparqlEndpoint
        <https://vocabs.ilc4clarin.ilc.cnr.it/graphdb10/repositories/h2iosc_skosmos> ;
    skosmos:sparqlGraph
        <https://vocabs.ilc4clarin.ilc.cnr.it/vocabularies/pl_20_500_11752_OPEN_1014/> .
```

This template is reused for all vocabularies managed by the pipeline. The configuration block mirrors the URI pattern already adopted for publishing the RDF data: each lexicon is stored in its own named graph (`https://vocabs.ilc4clarin.ilc.cnr.it/vocabularies/pl_{`*vocab_name*`}/`), and the same URI is referenced in `skosmos:sparqlGraph`, ensuring that SKOSMOS directs all SPARQL queries to the appropriate dataset. The descriptive fields then supply the interface with meanin gful labels, indicate the URI namespace for concepts, define the available languages, specify the default language, and connect the vocabulary to the shared SPARQL endpoint.

To group all Pan-Latin lexicons under a dedicated section in the interface, a custom category is introduced:

```
@prefix panlatin: <http://vocabs.ilc4clarin.ilc.cnr.it/panlatinlexicon> .

panlatin:LexiconCat a skos:Concept ;
    skos:prefLabel "REALITER - OTPL Pan-Latin Lexicons"@it,
                   "REALITER - OTPL Pan-Latin Lexicons"@en ;
    skos:definition
        "Category for grouping the Pan-Latin lexicons section"@en .
```

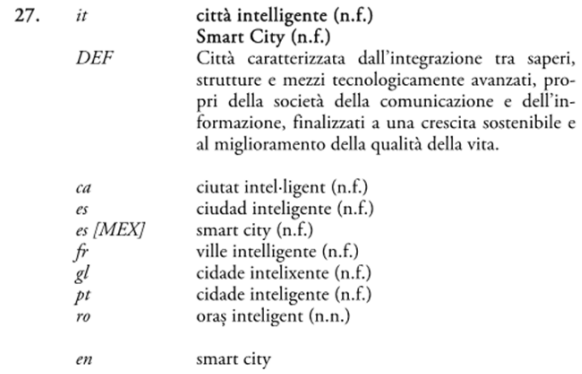
Each vocabulary is linked to this category via the `dc:subject` property, enabling SKOSMOS to display them together as a coherent thematic group.

3.6.2 Automatic Generation of Vocabulary Blocks

To avoid maintaining configuration blocks manually and to keep SKOSMOS aligned with the contents of the repository, all vocabulary declarations are produced automatically from the metadata and intermediate files generated by the pipeline. The script responsible for this task analyses the concept-scheme files and the corresponding structured data in JSON format, derives the vocabulary identifier, retrieves the title defined in the concept scheme, identifies all languages present in the labels and notes, and assembles a complete configuration block following the structure illustrated above. The resulting Turtle file (for example, `vocabs_data.ttl`) contains one declaration per vocabulary. Since its prefix declarations match those already present in the main SKOSMOS configuration, the operator simply appends the generated vocabulary blocks to `config.ttl`. After SKOSMOS is reloaded, each Pan-Latin lexicon becomes immediately available for browsing, consistently linked to its GraphDB named graph and integrated within the URI design defined earlier in the workflow.

4 Case Studies: REALITER Pan-Latin lexicons

This methodology has been applied in a use case involving the processing of specialised vocabularies (Grimaldi & Zanola, 2021) developed within the Pan-Latin Terminology Network - REALITER. This association, created in 1993, aims at promoting plurilingual communication in specialised fields. REALITER brings together experts, universities, associations, and institutions. Its institutional members include the Délégation générale à la langue française et aux langues de France, the Office québécois de la langue française, the Centre de Terminologia TERMCAT, the Servizo de Normalización Lingüística of



27.	*it*	**città intelligente (n.f.)** **Smart City (n.f.)**
	DEF	Città caratterizzata dall'integrazione tra saperi, strutture e mezzi tecnologicamente avanzati, propri della società della comunicazione e dell'informazione, finalizzati a una crescita sostenibile e al miglioramento della qualità della vita.
	ca	ciutat intel·ligent (n.f.)
	es	ciudad inteligente (n.f.)
	es [MEX]	smart city (n.f.)
	fr	ville intelligente (n.f.)
	gl	cidade intelixente (n.f.)
	pt	cidade inteligente (n.f.)
	ro	oraș inteligent (n.n.)
	en	smart city

Figure 4: Città intelligente, *Pan-Latin Smart City Lexicon* (2018), p. 21.

the Universidade de Santiago de Compostela, the Associazione Italiana per la Terminologia (Ass.I.Term) and the Osservatorio di Terminologie e Politiche Linguistiche of the Università Cattolica del Sacro Cuore [16]. REALITER encourages activities aimed at harmonising Romance languages and fostering linguistic diversity. More specifically, it develops and manages multilingual terminological resources to support experts, citizens, and academic communities (Gilardoni, 2011; Zanola, 2014). The Network members participate in collaborative projects to develop terminology products (e.g. lexicons) on domains of current interest. Since its foundation, around 40 lexicons across domains such as economics, environment, medicine, biology, digital technologies, education and fashion have been produced. The lexicons are multilingual since they show equivalents in seven Romance languages (Catalan, Spanish, French, Galician, Italian, Portuguese, Romanian), including some varieties such as Spanish spoken in Argentina, Colombia, Mexico; French spoken in Belgium, Canada; Brazilian Portuguese and Moldovan Romanian, as well as in English.

The development of REALITER multilingual lexicons follows the Common methodological principles [17]. Each member can propose a multilingual terminological project, that needs to be approved at the General Assembly of members. The proposals are evaluated according to the relevance of the chosen domain and its social interest. The member leading the project is responsible for coordinating the work and for the nomenclature proposal. For each concept, a definition in the reference Romance language is provided alongside with one or several terms used to designate it. Furthermore, one or more English equivalents are also identified to facilitate the communication among plurilingual working groups. For each language and variety considered, a team of terminologists and domain experts is gathered to provide the equivalents taking into account terminological variation and cultural aspects of each language and variety (Grimaldi et al., 2022; Lo Presti & Dankova, 2021). The structure and the content of lexicons can be slightly different from one project to another in order to meet specific needs of the domain of interest. For instance, the lexicons dealing with the new Olympic sports published in 2023-2024 [18] – beach volley, surf, trampoline and climbing – are divided in several thematic sections, such as equipment, competition types and athlete roles. Another interesting example concerns the definition of concepts: usually, the definition is given in the source language of the project (Fig. 4), however, in some cases, a definition is missing (Fig. 5) or it can be replaced by an image, as in the *Pan-Latin Lexicon of Collars and Sleeves in Fashion and Costume* (Fig. 6).

[16]For more information, see (Gilardoni, 2011; Zanola, 2014) and the REALITER website: https://www.realiter.net/ (accessed on 9 Decembrer 2025).

[17]For more information, please refer to the *Principes methodologiques du travail terminologique*. Available at: https://www.realiter.net/fr/quest-ce-que-realiter (accessed 9 Decembre 2025).

[18]These lexicons are available on the website : https://www.culture.gouv.fr/thematiques/langue-francaise-et-langues-de-france/agir-pour-les-langues/moderniser-et-enrichir-la-langue-francaise/nos-publications/vocabulaires-panlatins-du-sport (accessed 9 Decembre 2025).

madre biológica (n.f.)

cat	**mare biològica (n.f.)**
fra	**mère biologique (n.f.)**
glg	**nai biolóxica (s.f.)**
ita	**madre biologica (n.f.)**
por	**mãe biológica (s.f.)**
ron	**mamă biologică (s.f.)**
eng	***biological mother***

Figure 5: madre biológica, *Panlatin Bioethics Glossary* (2007), p. 35.

16. *it* **manica a sbuffo (s.f.)**

ca	màniga de fanal (n.f.)
	màniga de pernil (n.f.)
es	manga abullonada (s.f.)
es [MEX]	manga inflada (s.f.)
fr	manche gonflée (n.f.)
pt	manga curta bufante (s.f.)
pt [BR]	manga sopro (s.f.)
en	puffed sleeve
	balloon sleeve

Figure 6: Manica a sbuffo, *Pan-Latin Lexicon of Collars and Sleeves in Fashion and Costume* (2022), p. 25.

This case study addresses the conversion of 18 multilingual REALITER lexicons (3,467 entries) into SKOS and their hosting on the LLOD pilot SKOSMOS platform. The vocabularies, mainly in CSV format, contain labels, definitions and notes. Each entry contains a preferred term (`prefLabel`) and often several alternative terms (`altLabel`), with grammatical information following language-specific conventions (e.g. *s. m.* for masculine singular nouns in Romanian, *n. f.* for feminine singular nouns in French). Terms are marked with language and, for geographical variants, with country (e.g. Spanish (Mexico), Portuguese (Brazil)). Additional information is provided in notes. The collaborative nature of the data required dedicated parsing algorithms. By analysing the syntax of these taxonomies, an algorithmic foundation was defined to guide the conversion workflow. After conversion, the resources were integrated into an institutional vocabulary platform based on SKOSMOS, enabling browsing, multilingual search, and REST API access.

The Figure 7 represents the lexical entry *città intelligente* (smart city) of the *Pan-Latin Smart City Lexicon* in the SKOSMOS vocabulary platform.

5 Conclusions and Future Work

The methodology proposed in this article provides a structured approach for transforming linguistic data into SKOS resources, using Python notebooks to support data conversion and training. The pipeline integrates data acquisition, transformation, and publication, demonstrating its effectiveness in generating high-quality LLOD resources. The integration of these resources into a SKOSMOS-based platform enhances their accessibility and practical utility. Applying this methodology to Pan-Latin vocabularies has shown its ability to handle complex, multilingual data. Integrating these resources into the institutional vocabulary platform increases their visibility and supports reuse within the Linked Open Data community, both through a web interface and REST APIs.

Future developments will include adopting the Mapheator tool[19] to simplify mapping rule creation in

[19]https://github.com/oeg-upm/mapeathor

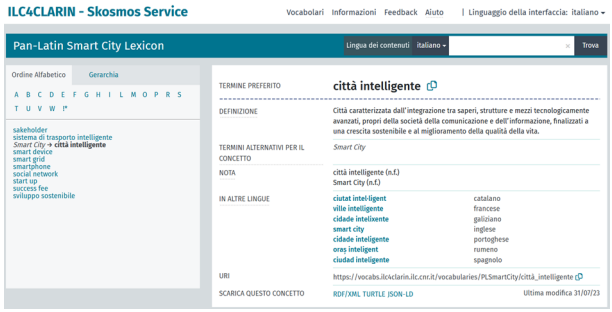


Figure 7: Città intelligente, *Pan-Latin Smart City Lexicon*.

formats like R2RML, RML, and YARRRML. The spreadsheet-based approach will make mapping more accessible for non-expert users and streamline the transformation process. The training materials from this project will be integrated into the H2IOSC training platform[20], which is currently being developed to support users in replicating the workflow and adopting LLOD best practices.

References

Berners-Lee, T. (2006). Linked open data design issues. https://www.w3.org/DesignIssues/LinkedData.html

Fiorelli, M., & Stellato, A. (2021). Lifting tabular data to RDF: A survey. In *Metadata and semantic research* (pp. 85–96). Springer International Publishing.

Gilardoni, S. (2011). I lessici della Rete panlatina di terminologia. In *Terminologie specialistiche e prodotti terminologici* (pp. 101–112). EDUCatt.

Grimaldi, C., Marzi, E., Puccini, P., Zanola, M. T., Zollo, S., et al. (2022). *Terminologia e interculturalità. problematiche e prospettive*. I libri di Emil.

Grimaldi, C., & Zanola, M. T. (2021). *Terminologie e vocabolari: Lessici specialistici e tesauri, glossari e dizionari*. Firenze University Press.

Khan, A. F., Chiarcos, C., Declerck, T., Gifu, D., García, E. G.-B., Gracia, J., Ionov, M., Labropoulou, P., Mambrini, F., McCrae, J. P., Pagé-Perron, É., Passarotti, M., Muñoz, S. R., & Truică, C.-O. (2022). When linguistics meets web technologies. recent advances in modelling linguistic linked data (P. Cimiano, J. Bosque-Gil, P. Cimiano, & M. Dojchinovski, Eds.). *Semantic Web*, *13*(6), 987–1050. https://doi.org/10.3233/sw-222859

Lamprecht, A.-L., Garcia, L., Kuzak, M., Martinez, C., Arcila, R., Martin Del Pico, E., Dominguez Del Angel, V., Van De Sandt, S., Ison, J., Martinez, P. A., et al. (2020). Towards fair principles for research software. *Data Science*, *3*(1), 37–59.

Lo Presti, M. V., & Dankova, K. (2021). Trattamento della terminologia culturale in una prospettiva multilingue. Il caso del *Lexique panlatin de la mobilité étudiante*. *AIDAinformazioni*, *39*(3-4), 45–66.

Zanola, M. T. (2014). Le réseau Realiter, un acteur du plurilinguisme. *PLAISANCE*, *11*(32), 149–165.

[20] https://h2iosc-training-platform.ilc4clarin.ilc.cnr.it/