

AI Systems Ontology: Relazione Progettuale

Autore:
Leonardo Magliolo

Corso di:
Semantic Web 2024/2025

Università di Torino - Facoltà di Informatica

1 Motivazioni

L'ontologia progettata per i sistemi di intelligenza artificiale costituisce uno strumento di ricerca avanzato per la rappresentazione della conoscenza, essenziale in ambito accademico e applicativo. La sua formalizzazione in OWL/SKOS permette di modellare in modo non ambiguo concetti chiave – ad esempio “modello”, “metodo di apprendimento”, “dominio applicativo” – e le relazioni fra essi, agevolando l'analisi comparativa di pubblicazioni scientifiche e dataset sperimentali.

Dal punto di vista della ricerca, l'ontologia abilita:

- Competency questions formalizzate e traducibili in query SPARQL ottimizzate, garantendo riproducibilità sperimentale e copertura del dominio;
- Inferenza automatica tramite reasoner OWL, che mette in luce relazioni implicite fra entità, supportando la scoperta di pattern non immediatamente evidenti nei dati di partenza.
- Ricerca semantica di informazioni: anziché limitarsi a stringhe testuali, i ricercatori possono interrogare il grafo con criteri di similarità concettuale o percorsi semantici, migliorando la pertinenza dei risultati e riducendo il rumore informativo.

Inoltre, l'ontologia è progettata per integrarsi con metodi NLP, consentendo di arricchire automaticamente il knowledge graph con entità e relazioni estratte da testi scientifici. Ciò abilita workflow ibridi in cui output di moduli NLP vengono mappati direttamente in triple RDF, accelerando la costruzione e l'aggiornamento della base di conoscenza.

Infine, l'allineamento con ontologie di dominio assicura la riusabilità e l'interoperabilità: gruppi di ricerca differenti possono estendere o integrare nuove conoscenze senza ricostruire da zero la struttura concettuale, favorendo collaborazioni e benchmarking su dataset comuni.

In sintesi, l'ontologia AI-oriented proposta costituisce un'infrastruttura di ricerca rigorosa, in cui formalismi ontologici, inferenza semantica e integrazione NLP concorrono a potenziare sia la qualità delle analisi bibliometriche sia la scoperta di nuove correlazioni scientifiche.

2 Requirements per la creazione dell'ontologia

2.1 Finalità generali

L'ontologia “AI Systems Ontology” è stata concepita per fornire una **rappresentazione semantica** rigorosa delle componenti chiave dei sistemi di intelligenza artificiale (paper, modelli, task, dataset, metodi, metriche), riducendo l'ambiguità terminologica e rendendo esplicite le relazioni formali tra entità. Tale codifica formale — realizzata in OWL/SKOS con namespace <http://example.org/ai-ontology#> — assicura:

- **Espressività** nel descrivere classi complesse (p. es. `DeepLearningModel`, `HybridModel`) e le loro proprietà strutturali e logiche;
- **Inferibilità**, grazie all'applicazione di un reasoner OWL, per estrarre relazioni implicite (p. es. implicazioni tra metodi e metriche di valutazione);
- **Interoperabilità**, mediante allineamenti (`owl:equivalentClass`, `rdfs:subClassOf`) con ontologie di settore.

2.2 Task specifici e contesto d'uso

L'ontologia è orientata a supportare principalmente tre task di ricerca e validazione:

1. **Consultazione semantica** – interrogazione di knowledge graph via SPARQL per recuperare informazioni strutturate (es. elenco di paper che “employModel” con un certo “TrainingAlgorithm”).
2. **Reference e verifica** – validazione delle assunzioni sperimentali attraverso competency questions tradotte in query SPARQL riproducibili (es. conteggio delle istanze di `SymbolicModel` e verifica dell'allineamento con risorse esterne).
3. **Estrazione e popolamento** – integrazione automatica di entità e relazioni estratte da testi scientifici (moduli NLP per Named Entity Recognition e Relation Extraction) per aggiornare dinamicamente la A-Box del grafo.

Il contesto di applicazione è un ambiente di ricerca accademica e industriale, in cui data scientist e knowledge engineer necessitano di un'infrastruttura semantica per analisi comparativa, audit dei workflow e produzione di report standardizzati.

2.3 Tipologia di utenti

- **Ricercatori in intelligenza artificiale:** necessitano di strumenti per formulare ipotesi, eseguire query complesse e confrontare approcci metodologici.
- **Knowledge engineer e ontologi:** hanno bisogno di estendere, allineare e validare strutture ontologiche secondo best practice OWL.
- **Data curator e bibliometrici:** utilizzano la codifica semantica per generare metriche di performance, visualizzazioni di trend e supportare processi di peer review e compliance normativa.

3 Descrizione del dominio

L'ontologia “AI Systems Ontology” è dedicata alla rappresentazione semantica del panorama della ricerca in Intelligenza Artificiale, con un focus sui principali artefatti scientifici (paper, autori, conferenze/riviste) e sugli elementi tecnici (task, metodi, modelli, dataset, metriche di valutazione) che ne costituiscono il contesto. In particolare, il dominio copre:

- Paper e loro metadati (titolo, data di pubblicazione, autori, venue) come unità di conoscenza primaria;
- Autori e venue (conferenze e riviste) per tracciare network di collaborazione e standard di divulgazione;
- Task (ad es. Computer Vision, Natural Language Processing), Method (classiche vs. deep learning), Model (simbolici, ibridi, reti neurali) e relativi `TrainingAlgorithm`, per descrivere le tecniche applicate;
- Dataset e Evaluation metrics (accuracy, F1-score, BLEU, ecc.) per caratterizzare la sperimentazione e i benchmark di performance.

Questa struttura semantica scaturisce dall'analisi di tassonomie esistenti come l'InnoGraph AI Taxonomy, che integra dati da Wikipedia, ACM CCS, CSO e altre fonti per offrire una visione olistica dei topic AI. Il dominio è stato ulteriormente raffinato tramite allineamenti a ontologie di settore (ad es. CSO per le aree di ricerca) e pattern di modellazione OWL che garantiscono coerenza logica e inferibilità.

La “The InnoGraph Artificial Intelligence Taxonomy” è stata utilizzata come punto di riferimento centrale nella progettazione dell’“AI Systems Ontology”. In particolare, ha fornito l'ispirazione per:

Definire l'ampiezza del dominio, includendo metodi, modelli, task, dataset, metriche, pubblicazioni, autori e application areas.

Integrare più fonti esterne attraverso annotazioni SKOS, in modo da garantire interoperabilità con CSO, Wikipedia/CPC e altre tassonomie consolidate.

Segmentare metodi e modelli secondo la simili suddivisione (simbolici, ML tradizionale, deep learning) e gestire dataset e valutazioni con proprietà inverse, replicando lo schema di InnoGraph.

In sintesi, l'approccio di InnoGraph—ampio, multi-sorgente e semantico—ha guidato ogni decisione di design, rendendo la tassonomia un'ispirazione costante per la struttura e i contenuti dell'ontologia.

4 Competency Questions

- **Recupero direzionale**

- Quali entità (paper, modelli, autori, task, ecc.) soddisfano una data relazione semantica (ad esempio: “impiegano un certo tipo di modello” o “sono pubblicati in una conferenza”)?

- **Analisi quantitativa**

- Qual è la distribuzione numerica di entità raggruppate per una proprietà di interesse (ad esempio: numero di paper per task, numero di paper per anno, numero di autori per venue)?

- **Verifica di copertura**

- Quali entità definite nel dominio non risultano collegate ad altre entità rilevanti (ad esempio: dataset non utilizzati, modelli non impiegati, task non affrontati)?

- **Pattern compositi**

- Quali combinazioni di classi e proprietà ricorrono frequentemente nei documenti o nei grafi sperimentali (ad esempio: coppie modello–algoritmo di training, coppie metodo–metrica di valutazione)?

- **Allineamento esterno**

- Come si confrontano e si integrano le entità interne (paper, autori, modelli, dataset) con risorse esterne di riferimento (ad esempio: conferenze, repository)?

5 Documentazione sul dominio

5.1 Struttura concettuale del dominio

Il dominio della ricerca in AI può essere descritto come una galassia di entità e relazioni che riflettono il ciclo di vita dell'innovazione scientifica. Gli elementi fondamentali di questo ecosistema includono:

- **Persone:** Ricercatori, autori di articoli, inventori di brevetti, sviluppatori di software, docenti universitari e leader di progetti.
- **Organizzazioni:** Università, centri di ricerca, aziende, startup, agenzie di finanziamento, consorzi e associazioni professionali.
- **Risorse scientifiche:** Articoli, capitoli di libro, tesi, presentazioni a conferenze, dataset e modelli AI pubblicati.
- **Tecniche, modelli e algoritmi:** Algoritmi, architetture, paradigmi di apprendimento (come machine learning, deep learning, reinforcement learning), metodi specifici (es. attenzione, regolarizzazione), modelli di stato dell'arte (es. GPT-4, BERT, EfficientNet, ResNet).
- **Task e problemi:** Problemi di classificazione, regressione, generazione di testo, riconoscimento di immagini, segmentazione, question answering, ecc.
- **Ambiti applicativi:** Domini verticali di applicazione come medicina, agricoltura di precisione, finanza, industria 4.0, trasporti autonomi, energia, legaltech, robotica, ecc.
- **Software e infrastrutture:** Framework (TensorFlow, PyTorch, Keras), librerie di supporto, piattaforme di condivisione di modelli (HuggingFace), ambienti di calcolo distribuito.
- **Valutazione e benchmarking:** Dataset di test, leaderboards, competizioni, metriche standardizzate di performance e robustezza.

5.2 Esempio concreto di entità e relazioni

Un esempio reale che illustra la complessità del dominio è l'articolo scientifico pubblicato su arXiv, associato a un modello disponibile su GitHub, validato tramite un benchmark su un dataset pubblico (ad esempio ImageNet), e citato in una challenge (come la ImageNet Large Scale Visual Recognition Challenge). In questo esempio, si evidenziano le seguenti relazioni:

- L'autore è affiliato a una università e ha rilasciato il codice su GitHub.
- Il modello è stato addestrato su un dataset pubblico e i risultati sono confrontati in una leaderboard internazionale.
- L'articolo è referenziato da altri lavori e può essere oggetto di brevetto o di ulteriore sviluppo da parte di aziende.

5.3 Fonti documentali e tassonomie di riferimento

- Wikipedia e DBpedia: Articoli, categorie e glossari che coprono tecniche, applicazioni, dataset, personalità di rilievo e storie dell'AI. Gli outline e le liste tematiche aiutano a navigare rapidamente fra le principali aree dell'AI.
- arXiv: Portale di preprint dove la quasi totalità della letteratura più recente viene pubblicata, con categorizzazioni tematiche (cs.AI, cs.LG, cs.CV, ecc.) che riflettono la suddivisione del dominio scientifico.
- PapersWithCode: Portale che collega pubblicazioni, codice sorgente, dataset e risultati di benchmarking.
- ACM CCS, Computer Science Ontology (CSO): Sistemi di classificazione e tassonomie che riflettono la struttura della ricerca in computer science e AI.
- Patent Classification Systems (es. Cooperative Patent Classification, CPC): Sistemi di classificazione che inquadrano le invenzioni AI rispetto ai loro ambiti applicativi industriali e tecnologici.

Esempi di schemi e dati reali:

Infobox di Wikipedia

```
| Nome modello      = GPT-4
| Tipo              = Large Language Model
| Sviluppatore      = OpenAI
| Anno              = 2023
| Applicazioni      = Generazione di testo, traduzione automatica, question answering
| Dataset di training = Common Crawl, Wikipedia, BooksCorpus
```

Dataset PapersWithCode

Task: Image Classification

Dataset: ImageNet

Model: ResNet-50

Score: Top-1 Accuracy 76.0%

Framework: PyTorch

Paper: "Deep Residual Learning for Image Recognition"

Repository: <https://github.com/KaimingHe/deep-residual-networks>

Estratto da arXiv (cs.AI)

Titolo: "A General Language Assistant as a Laboratory for Alignment"

Autori: John Smith, Jane Doe

Categorie: cs.AI, cs.CL

Abstract: "We introduce a new framework for language model alignment con human prefer

Link: <https://arxiv.org/abs/2302.12345>

Tabella di classi di problemi (esempio schematico)

Macroarea	Sottocategoria	Esempio Task	Dataset tipico
Computer Vision	Image Classification	Identificazione oggetti	ImageNet, CIFAR
Natural Language	Question Answering	Risposta a domande libere	SQuAD, Natural
Robotics	Motion Planning	Navigazione robotica	KITTI, Waymo
Bioinformatics	Protein Structure	Predizione struttura proteica	AlphaFold

5.4 Standard e modelli di riferimento

Il dominio fa costantemente riferimento a una pluralità di tassonomie, classificazioni e sistemi di reference che servono a garantire interoperabilità, comparabilità dei risultati e chiarezza terminologica. Esempi rilevanti:

- ACM Computing Classification System: Classifica la letteratura scientifica secondo aree tematiche codificate.
- CSO (Computer Science Ontology): Tassonomia aggiornata automaticamente dalla letteratura scientifica.
- CPC (Cooperative Patent Classification): Classificazione dettagliata delle invenzioni brevettate in base alla loro natura tecnologica e applicativa.
- EuroSciVoc: Thesaurus europeo che classifica i progetti di ricerca secondo aree scientifiche e ingegneristiche, incluso l'AI.

Questi sistemi permettono di mappare le entità del dominio in modo coerente, facilitando l'allineamento e la federazione di dati fra fonti eterogenee.

5.5 Considerazioni sulla varietà e profondità del dominio

Il dominio della ricerca scientifica in AI si caratterizza per una notevole eterogeneità: i confini fra sottodiscipline sono sfumati e spesso emergono nuove aree interdisciplinari (es. AI per le neuroscienze, AI etica, explainable AI, AI generativa). Allo stesso tempo, la profondità delle specializzazioni consente una verticalizzazione molto marcata: dalla teoria alla pratica, dall'algoritmo generale al micro-task su dati specifici.

L'evoluzione continua delle metodologie, l'emergere di nuovi paradigmi (es. foundation models, few-shot learning, AI ibrida simbolica-connessionista) e l'intensa produzione scientifica richiedono strumenti strutturati e condivisi per rappresentare, organizzare e ricercare la conoscenza del dominio.

5.6 Esempio reale per la A-box

Caso d'uso: Un utente vuole scoprire quali sono i modelli di AI più utilizzati in ambito medico per la diagnostica su immagini radiologiche.

Esempio di dato:

- Task: Medical Image Classification
- Dataset: ChestX-ray14
- Modello: DenseNet-121
- Autore principale: Dr. Jane Doe
- Organizzazione: Stanford University
- Articolo di riferimento: "CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning"
- Score: AUC 0.84

Questa scheda, modellata come individuo nella A-box dell'ontologia, riflette la varietà delle entità e delle relazioni che compongono il dominio, e la necessità di mettere in relazione task, dataset, modelli, autori, organizzazioni e risultati.

6 Ontology Pitfall Scanner (OOPS)

OOPS riporta checklist vuota

7 Visualizzazione

Di seguito vengono presentate tre visualizzazioni chiave che illustrano rispettivamente la tassonomia delle classi, la struttura relazionale attorno alla classe `Model` e la struttura relazionale attorno alla classe `Paper`. Le immagini sono state salvate nella cartella `images`.

7.1 Tassonomia delle classi

La Figura 1 mostra la gerarchia delle classi a partire da `owl:Thing` come radice. In alto, le classi di dominio più generali (ad esempio `ApplicationArea`, `Author`, `Conference`, `Dataset`, `Evaluation`, `Journal`), seguite dalle sottoclassi relative a `Method` e `Model`. Nello specifico:

- `Method` è suddivisa in `HybridMethod`, `MachineLearningMethod` e `RuleBasedMethod`. `MachineLearningMethod` si articola ulteriormente in `DeepLearningMethod` e `TraditionalMLMethod`, mentre `RuleBasedMethod` ha come sottoclasse `SymbolicMethod`.
- `Model` presenta analoghe suddivisioni: `HybridModel`, `MachineLearningModel` e `RuleBasedModel`. `MachineLearningModel` si distingue in `DeepLearningModel` e `TraditionalMLModel`, mentre `RuleBasedModel` presenta la sottoclasse `SymbolicModel`.
- Le classi “foglia” includono anche `Paper`, `Repository`, `Task` e `TrainingAlgorithm`.

Questa tassonomia aiuta a comprendere come le entità principali siano organizzate in aree tematiche coerenti e gerarchiche, facilitando l'uso di ragionatori OWL per l'inferenza sui sottoinsiemi e la coerenza logica.

7.2 Struttura relazionale della classe `Model`

Nella Figura 2 è rappresentata la classe `Model` al centro del grafo, con:

- I nodi corrispondenti alle sottoclassi dirette (`HybridModel`, `MachineLearningModel`, `RuleBasedModel`), ciascuna a sua volta collegata alle proprie sottoclassi (`DeepLearningModel`, `TraditionalMLModel`, `SymbolicModel`).
- Le proprietà oggetto principali: ad esempio, `employsModel` (utilizzata per collegare un `Paper` a un determinato `Model`) e la proprietà inversa `isModelEmployedBy`.
- Le relazioni di disgiunzione (annotate come `owl:disjointWith`) che separano classi semanticamente incompatibili (ad esempio, `Model` è disgiunta da `Conference`, `Dataset`, `Evaluation`, `Journal`, `Paper`, `Task`, `Author`, `Repository`, `Method`, `Thing` e `Nothing`).

Questa visualizzazione evidenzia come **Model** si collochi all'interno dell'ontologia, mostrando sia l'albero di sottoclassi che le principali proprietà relazionali con le altre classi di dominio.

7.3 Struttura relazionale della classe **Paper**

La Figura 3 mostra la classe **Paper** al centro, con:

- Alcune istanze di esempio (ad es. **Paper1**, **Paper2**, **Paper3**, ...) collegate tramite l'arco `rdf:type` alla classe **Paper**.
- Le relazioni verso le classi di dominio (quali **Author**, **Model**, **Task**, **Conference**, **Dataset**, **Evaluation**, **Method**), ciascuna marcata come `owl:disjointWith` rispetto a **Paper**. Ciò sottolinea come un'istanza di **Paper** non possa appartenere simultaneamente a nessuna di queste altre classi.
- Le frecce etichettate con `predicate` e `type` per illustrare le proprietà usate (ad esempio, `hasTitle`, `hasAuthor`, `addressesTask`, `employsModel`, `publishedIn`, `hasDate`, `usesDataset`, `usesEvaluationMetric`).

Questa mappa relazionale permette di comprendere rapidamente come gli articoli scientifici (**Paper**) si colleghino a tutti gli altri elementi dell'ontologia, nonché quali proprietà semanticamente impattano la coerenza del grafo (tramite disjointness).

Subject	Predicate	Object
:Author_AliceSmith	a	:Author
:Author_AliceSmith	:hasTitle	"Alice Smith"
:Author_BobRossi	a	:Author
:Author_BobRossi	:hasTitle	"Bob Rossi"
:ComputerVisionClass	a	:ApplicationArea
:ComputerVisionClass	rdfs:label	"Computer Vision"
:Task_ImageClassification	a	:Task
:Task_ImageClassification	:hasTitle	"Image Classification"
:Task_ImageClassification	:hasApplicationArea	:ComputerVisionClass
:SGD	a	:TrainingAlgorithm
:SGD	:hasTitle	"SGD"
:ResNet50	a	:DeepLearningModel
:ResNet50	:usesTrainingAlgorithm	:SGD
:Paper1	a	:Paper
:Paper1	:hasTitle	"Efficient ResNet50 for Image Classification"
:Paper1	:hasAuthor	:Author_AliceSmith, :Author_BobRossi
:Paper1	:publishedIn	:CVPR2023
:Paper1	:hasDate	"2023-06-20"sd:date
:Paper1	:addressesTask	:Task_ImageClassification
:Paper1	:employsModel	:ResNet50
:Paper1	:usesDataset	:ImageNet
:Paper1	:hasEvaluation	:Accuracy, :Precision, :Recall

Table 1: Esempio compatto di triple RDF rappresentate a livello dell'ontologia

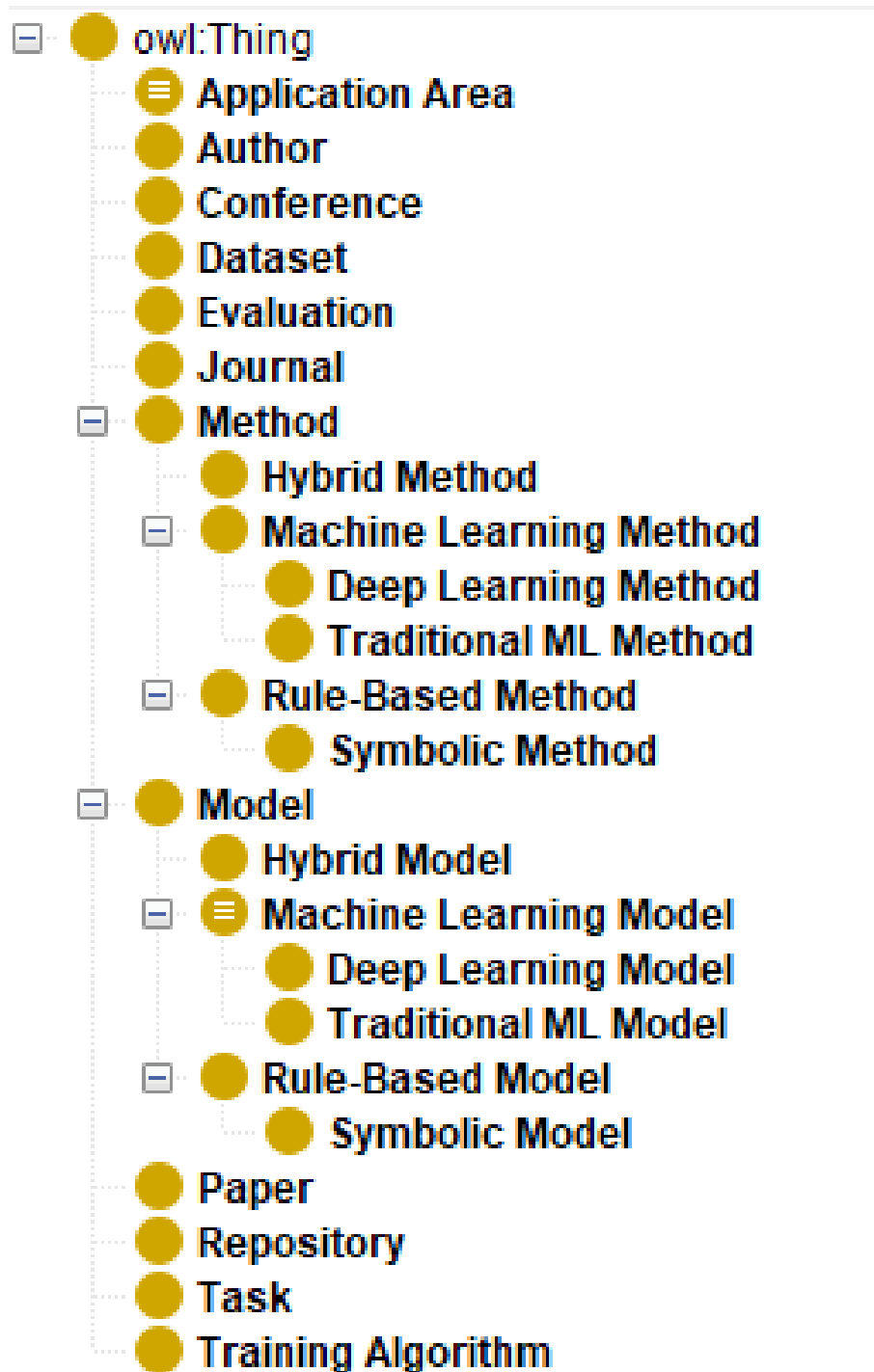
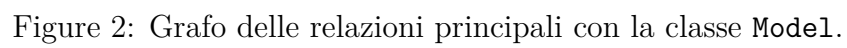


Figure 1: Tassonomia delle classi principali nell'ontologia AI Systems Ontology.



8 L'ontologia in formato Turtle

```
@prefix : <http://example.org/ai-ontology#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix vann: <http://purl.org/vann/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@base <http://example.org/ai-ontology#> .
```

```
<http://example.org/ai-ontology> rdf:type owl:Ontology ;
                                dc:creator "Leonardo Magliolo" ;
                                dc:date "2024-03-25" ;
                                dc:description "Ontology modeling AI systems, methods, and tools" ;
                                dc:rights "CC BY 4.0" ;
                                dc:title "Ontology of AI Systems"@en ;
                                dcterms:license <https://creativecommons.org/licenses/by/4.0/> ;
                                vann:preferredNamespacePrefix "ai" ;
                                vann:preferredNamespaceUri "http://example.org/ai-ontology#" ;
                                rdfs:label "AI Systems Ontology"@en .
```

```
#####
#   Annotation properties
#####
```

```
### http://purl.org/dc/elements/1.1/creator
dc:creator rdf:type owl:AnnotationProperty .
```

```
### http://purl.org/dc/elements/1.1/date
dc:date rdf:type owl:AnnotationProperty .
```

```
### http://purl.org/dc/elements/1.1/description
dc:description rdf:type owl:AnnotationProperty .
```

```
### http://purl.org/dc/elements/1.1/rights
dc:rights rdf:type owl:AnnotationProperty .
```

```
### http://purl.org/dc/elements/1.1/title
dc:title rdf:type owl:AnnotationProperty .
```

```
### http://purl.org/dc/terms/license
dcterms:license rdf:type owl:AnnotationProperty .
```

```
### http://purl.org/vann/preferredNamespacePrefix
vann:preferredNamespacePrefix rdf:type owl:AnnotationProperty .
```

```
### http://purl.org/vann/preferredNamespaceUri
vann:preferredNamespaceUri rdf:type owl:AnnotationProperty .
```

```
### http://www.w3.org/2004/02/skos/core#closeMatch
skos:closeMatch rdf:type owl:AnnotationProperty .
```

```
### http://www.w3.org/2004/02/skos/core#exactMatch
skos:exactMatch rdf:type owl:AnnotationProperty .
```

```
### http://www.w3.org/2004/02/skos/core#narrowMatch
skos:narrowMatch rdf:type owl:AnnotationProperty .
```

```
#####
#   Datatypes
#####
```

```
### http://www.w3.org/2001/XMLSchema#date
xsd:date rdf:type rdfs:Datatype .
```

```
#####
#   Object Properties
#####
```

```
### http://example.org/ai-ontology#addressesTask
:addressesTask rdf:type owl:ObjectProperty ;
               owl:inverseOf :isTaskAddressedBy ;
               rdfs:domain [ rdf:type owl:Class ;
                             owl:unionOf ( :Model
                                              :Paper
                                              )
                           ] ;
               rdfs:range :Task ;
               rdfs:comment "Relates a paper or model to the task it addresses."@en ;
               rdfs:label "addresses task"@en .
```

```

### http://example.org/ai-ontology#applicationAreaOf
:applicationAreaOf rdf:type owl:ObjectProperty ;
                    owl:inverseOf :hasApplicationArea ;
                    rdfs:domain :ApplicationArea ;
                    rdfs:range :Task ;
                    rdfs:comment "Relates an application area to the tasks that belong
                    rdfs:label "application area of"@en .

### http://example.org/ai-ontology#authored
:authored rdf:type owl:ObjectProperty ;
           owl:inverseOf :hasAuthor ;
           rdfs:domain :Author ;
           rdfs:range :Paper ;
           rdfs:comment "Relates an author to the papers they wrote. (Inverse of hasAuthor)"@en ;
           rdfs:label "authored"@en .

### http://example.org/ai-ontology#coAuthorOf
:coAuthorOf rdf:type owl:ObjectProperty ,
              owl:SymmetricProperty ;
            rdfs:domain :Author ;
            rdfs:range :Author ;
            owl:propertyChainAxiom ( :authored
                                       :hasAuthor
                                       ) ;
            rdfs:comment "Two authors are co-authors if they have written the same paper" ;
            rdfs:label "co-author of"@en .

### http://example.org/ai-ontology#employsModel
:employsModel rdf:type owl:ObjectProperty ;
              owl:inverseOf :isModelEmployedBy ;
              rdfs:domain :Paper ;
              rdfs:range :Model ;
              rdfs:comment "Relates a paper to the model it employs."@en ;
              rdfs:label "employs model"@en .

### http://example.org/ai-ontology#hasApplicationArea
:hasApplicationArea rdf:type owl:ObjectProperty ;
                    rdfs:domain :Task ;
                    rdfs:range :ApplicationArea ;
                    rdfs:comment "Relates a task to its application area."@en ;
                    rdfs:label "has application area"@en .

```



```

### http://example.org/ai-ontology#hasAuthor
:hasAuthor rdf:type owl:ObjectProperty ;
           rdfs:domain :Paper ;
           rdfs:range :Author ;
           rdfs:comment "Relates a paper to its author(s)."@en ;
           rdfs:label "has author"@en .

### http://example.org/ai-ontology#hasAuthorList
:hasAuthorList rdf:type owl:ObjectProperty ;
               owl:inverseOf :isAuthorListOf ;
               rdfs:domain :Paper ;
               rdfs:range :AuthorList ;
               rdfs:comment "Relates a paper to its ordered list of authors."@en ;
               rdfs:label "has author list"@en .

### http://example.org/ai-ontology#hasEvaluation
:hasEvaluation rdf:type owl:ObjectProperty ;
               owl:inverseOf :isEvaluationOf ;
               rdfs:domain [ rdf:type owl:Class ;
                             owl:unionOf ( :Model
                                              :Paper
                                              )
                           ] ;
               rdfs:range :Evaluation ;
               rdfs:comment "Relates a paper or model to its evaluation metric."@en ;
               rdfs:label "has evaluation"@en .

### http://example.org/ai-ontology#hasFirstNode
:hasFirstNode rdf:type owl:ObjectProperty ;
               owl:inverseOf :isFirstNodeOf ;
               rdfs:domain :AuthorList ;
               rdfs:range :AuthorListNode ;
               rdfs:comment "Relates an author list to its first node."@en ;
               rdfs:label "has first node"@en .

### http://example.org/ai-ontology#hasListAuthor
:hasListAuthor rdf:type owl:ObjectProperty ;
               owl:inverseOf :isListAuthorOf ;
               rdfs:domain :AuthorListNode ;
               rdfs:range :Author ;
               rdfs:comment "Relates a node in the list to the corresponding author."@en ;
               rdfs:label "has list author"@en .

```

```

### http://example.org/ai-ontology#hasMethod
:hasMethod rdf:type owl:ObjectProperty ;
            owl:inverseOf :isMethodOfModel ;
            rdfs:domain :Model ;
            rdfs:range :Method ;
            rdfs:comment "Relates a model to the method it implements."@en ;
            rdfs:label "has method"@en .

### http://example.org/ai-ontology#hasNextNode
:hasNextNode rdf:type owl:ObjectProperty ;
              owl:inverseOf :isPreviousNodeOf ;
              rdf:type owl:TransitiveProperty ;
              rdfs:domain :AuthorListNode ;
              rdfs:range :AuthorListNode ;
              rdfs:comment "Relates a node to the next node in the list."@en ;
              rdfs:label "has next node"@en .

### http://example.org/ai-ontology#hostsDataset
:hostsDataset rdf:type owl:ObjectProperty ;
               owl:inverseOf :isHostedBy ;
               rdfs:domain :Repository ;
               rdfs:range :Dataset ;
               rdfs:comment "Relates a repository to the datasets it hosts."@en ;
               rdfs:label "hosts dataset"@en .

### http://example.org/ai-ontology#isAuthorListOf
:isAuthorListOf rdf:type owl:ObjectProperty ;
                 rdfs:domain :AuthorList ;
                 rdfs:range :Paper ;
                 rdfs:comment "Relates an author list to the paper that owns it (inverse)."@en ;
                 rdfs:label "is author list of"@en .

### http://example.org/ai-ontology#isDatasetUsedBy
:isDatasetUsedBy rdf:type owl:ObjectProperty ;
                  owl:inverseOf :usesDataset ;
                  rdfs:domain :Dataset ;
                  rdfs:range [ rdf:type owl:Class ;
                               owl:unionOf ( :Model
                                                :Paper
                                              )
                             ] ;
                  rdfs:comment "Relates a dataset to the papers or models that rely on it."@en ;
                  rdfs:label "is dataset used by"@en .

```

```

### http://example.org/ai-ontology#isEvaluationOf
:isEvaluationOf rdf:type owl:ObjectProperty ;
                rdfs:domain :Evaluation ;
                rdfs:range [ rdf:type owl:Class ;
                             owl:unionOf ( :Model
                                              :Paper
                                              )
                             ] ;
                rdfs:comment "Relates an evaluation metric to the paper or model it as
                rdfs:label "is evaluation of"@en .

### http://example.org/ai-ontology#isFirstNodeOf
:isFirstNodeOf rdf:type owl:ObjectProperty ;
               rdfs:domain :AuthorListNode ;
               rdfs:range :AuthorList ;
               rdfs:comment "Relates an author list node to the list it is the first c
               rdfs:label "is first node of"@en .

### http://example.org/ai-ontology#isHostedBy
:isHostedBy rdf:type owl:ObjectProperty ;
            rdfs:domain :Dataset ;
            rdfs:range :Repository ;
            rdfs:comment "Relates a dataset to the repository that hosts it. (Inverse
            rdfs:label "is hosted by"@en .

### http://example.org/ai-ontology#isListAuthorOf
:isListAuthorOf rdf:type owl:ObjectProperty ;
                rdfs:domain :Author ;
                rdfs:range :AuthorListNode ;
                rdfs:comment "Relates a list node to the author it represents (inverse
                rdfs:label "is list author of"@en .

### http://example.org/ai-ontology#isMethodOfModel
:isMethodOfModel rdf:type owl:ObjectProperty ;
                 rdfs:domain :Method ;
                 rdfs:range :Model ;
                 rdfs:comment "Relates a method to the model that implements it. (Inv
                 rdfs:label "is method of model"@en .

### http://example.org/ai-ontology#isMethodUsedBy
:isMethodUsedBy rdf:type owl:ObjectProperty ;

```

```

owl:inverseOf :usesMethod ;
rdfs:domain :Method ;
rdfs:range [ rdf:type owl:Class ;
              owl:unionOf ( :Model
                              :Paper
                            )
            ] ;
rdfs:comment "Relates a method to the papers or models that employ it"
rdfs:label "is method used by"@en .

```

```

### http://example.org/ai-ontology#isModelEmployedBy
:isModelEmployedBy rdf:type owl:ObjectProperty ;
rdfs:domain :Model ;
rdfs:range :Paper ;
rdfs:comment "Relates a model to the papers that employ it. (Inverse)"
rdfs:label "is model employed by"@en .

```

```

### http://example.org/ai-ontology#isPreviousNodeOf
:isPreviousNodeOf rdf:type owl:ObjectProperty ,
                  owl:TransitiveProperty ;
rdfs:domain :AuthorListNode ;
rdfs:range :AuthorListNode ;
rdfs:comment "Relates a node to the node that precedes it in the author list"
rdfs:label "is previous node of"@en .

```

```

### http://example.org/ai-ontology#isTaskAddressedBy
:isTaskAddressedBy rdf:type owl:ObjectProperty ;
rdfs:domain :Task ;
rdfs:range [ rdf:type owl:Class ;
              owl:unionOf ( :Model
                              :Paper
                            )
            ] ;
rdfs:comment "Relates a task to the papers or models that address it"
rdfs:label "is task addressed by"@en .

```

```

### http://example.org/ai-ontology#isTrainingAlgorithmOf
:isTrainingAlgorithmOf rdf:type owl:ObjectProperty ;
owl:inverseOf :usesTrainingAlgorithm ;
rdfs:domain :TrainingAlgorithm ;
rdfs:range :Model ;
rdfs:comment "Relates a training algorithm to the model it is trained on"
rdfs:label "is training algorithm of"@en .

```

```

### http://example.org/ai-ontology#publishedIn
:publishedIn rdf:type owl:ObjectProperty ;
              owl:inverseOf :publishes ;
              rdf:type owl:FunctionalProperty ;
              rdfs:domain :Paper ;
              rdfs:range [ rdf:type owl:Class ;
                           owl:unionOf ( :Conference
                                           :Journal
                                           )
                           ] ;
              rdfs:comment "Relates a paper to its publication venue."@en ;
              rdfs:label "published in"@en .

### http://example.org/ai-ontology#publishes
:publishes rdf:type owl:ObjectProperty ;
           rdfs:domain [ rdf:type owl:Class ;
                        owl:unionOf ( :Conference
                                        :Journal
                                        )
                        ] ;
           rdfs:range :Paper ;
           rdfs:comment "Relates a conference or journal to the papers published with."@en ;
           rdfs:label "publishes"@en .

### http://example.org/ai-ontology#usesDataset
:usesDataset rdf:type owl:ObjectProperty ;
             rdfs:domain [ rdf:type owl:Class ;
                          owl:unionOf ( :Model
                                          :Paper
                                          )
                          ] ;
             rdfs:range :Dataset ;
             rdfs:comment "Relates a paper or model to the dataset it uses."@en ;
             rdfs:label "uses dataset"@en .

### http://example.org/ai-ontology#usesMethod
:usesMethod rdf:type owl:ObjectProperty ;
            rdfs:domain [ rdf:type owl:Class ;
                        owl:unionOf ( :Model
                                        :Paper
                                        )
                        ] ;
            rdfs:range :Method ;
            rdfs:comment "Relates a paper or model to the method it uses."@en ;

```

```

        rdfs:label "uses method"@en .

### http://example.org/ai-ontology#usesTrainingAlgorithm
:usesTrainingAlgorithm rdf:type owl:ObjectProperty ;
                        rdfs:domain :Model ;
                        rdfs:range :TrainingAlgorithm ;
                        rdfs:comment "Relates a model to the training algorithm it uses" ;
                        rdfs:label "uses training algorithm"@en .

#####
# Data properties
#####

### http://example.org/ai-ontology#hasDOI
:hasDOI rdf:type owl:DatatypeProperty ,
           owl:FunctionalProperty ;
        rdfs:domain :Paper ;
        rdfs:range xsd:string ;
        rdfs:comment "Each paper has at most one DOI."@en ;
        rdfs:label "has DOI"@en .

### http://example.org/ai-ontology#hasDate
:hasDate rdf:type owl:DatatypeProperty ,
           owl:FunctionalProperty ;
        rdfs:domain [ rdf:type owl:Class ;
                       owl:unionOf ( :Conference
                                       :Dataset
                                       :Evaluation
                                       :Journal
                                       :Paper
                                   )
                     ] ;
        rdfs:range xsd:date ;
        rdfs:comment "Date of publication/event."@en ;
        rdfs:label "has date"@en .

### http://example.org/ai-ontology#hasName
:hasName rdf:type owl:DatatypeProperty ;
        rdfs:domain :Author ;
        rdfs:range xsd:string ;
        rdfs:comment "Nome completo dell'autore."@en ;
        rdfs:label "has name"@en .

```

```

### http://example.org/ai-ontology#hasTitle
:hasTitle rdf:type owl:DatatypeProperty ,
            owl:FunctionalProperty ;
    rdfs:domain [ rdf:type owl:Class ;
                  owl:unionOf ( :Dataset
                                :Evaluation
                                :Method
                                :Model
                                :Paper
                                :Repository
                                :Task
                              )
                ] ;
    rdfs:range xsd:string ;
    rdfs:comment "Title of entity."@en ;
    rdfs:label "has title"@en .

#####
#    Classes
#####

### http://example.org/ai-ontology#ApplicationArea
:ApplicationArea rdf:type owl:Class ;
    owl:equivalentClass [ rdf:type owl:Class ;
                           owl:oneOf ( :ComputerVision
                                         :NaturalLanguageProcessing
                                       )
                         ] ;
    rdfs:comment """
        Enumerazione delle possibili aree applicative (es. CV, NLP).
    """@en ;
    rdfs:label "Application Area"@en .

### http://example.org/ai-ontology#Author
:Author rdf:type owl:Class ;
    rdfs:comment "Author of papers."@en ;
    rdfs:label "Author"@en ;
    skos:broadMatch <http://xmlns.com/foaf/0.1/Person> .

### http://example.org/ai-ontology#AuthorList
:AuthorList rdf:type owl:Class ;
    rdfs:comment "A list of authors in a specific order."@en ;
    rdfs:label "Author List"@en .

```

```

### http://example.org/ai-ontology#AuthorListNode
:AuthorListNode rdf:type owl:Class ;
    rdfs:comment "A node in the ordered author list."@en ;
    rdfs:label "Author List Node"@en .

### http://example.org/ai-ontology#ComputerVision
:ComputerVision rdf:type owl:Class ;
    rdfs:subClassOf :ApplicationArea .

### http://example.org/ai-ontology#Conference
:Conference rdf:type owl:Class ;
    rdfs:comment "Academic conference."@en ;
    rdfs:label "Conference"@en ;
    skos:exactMatch <http://swrc.ontoware.org/ontology#Conference> .

### http://example.org/ai-ontology#Dataset
:Dataset rdf:type owl:Class ;
    rdfs:subClassOf [ rdf:type owl:Restriction ;
        owl:onProperty :hasTitle ;
        owl:minQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
        owl:onDataRange xsd:string
    ] ;
    rdfs:comment "Dataset used in AI."@en ;
    rdfs:label "Dataset"@en ;
    skos:exactMatch <http://schema.org/Dataset> .

### http://example.org/ai-ontology#DeepLearningMethod
:DeepLearningMethod rdf:type owl:Class ;
    rdfs:subClassOf :MachineLearningMethod ;
    rdfs:comment "A method based on deep learning techniques."@en ;
    rdfs:label "Deep Learning Method"@en .

### http://example.org/ai-ontology#DeepLearningModel
:DeepLearningModel rdf:type owl:Class ;
    rdfs:subClassOf :MachineLearningModel ;
    rdfs:comment "Deep learning neural networks."@en ;
    rdfs:label "Deep Learning Model"@en ;
    skos:broadMatch <https://cso.kmi.open.ac.uk/topics/neural_networks> .

### http://example.org/ai-ontology#Evaluation
:Evaluation rdf:type owl:Class ;
    rdfs:comment "Evaluation metric."@en ;

```



```

        rdfs:label "Evaluation"@en .

### http://example.org/ai-ontology#HybridMethod
:HybridMethod rdf:type owl:Class ;
               rdfs:subClassOf :Method ;
               rdfs:comment "A method integrating symbolic and sub-symbolic approaches" ;
               rdfs:label "Hybrid Method"@en .

### http://example.org/ai-ontology#HybridModel
:HybridModel rdf:type owl:Class ;
              rdfs:subClassOf :Model ;
              rdfs:comment "Hybrid AI models integrating symbolic and sub-symbolic."@en ;
              rdfs:label "Hybrid Model"@en ;
              skos:exactMatch <https://cso.kmi.open.ac.uk/topics/neurosymbolic_ai> .

### http://example.org/ai-ontology#Journal
:Journal rdf:type owl:Class ;
          rdfs:comment "Academic journal."@en ;
          rdfs:label "Journal"@en ;
          skos:exactMatch <http://swrc.ontoware.org/ontology#Journal> .

### http://example.org/ai-ontology#MachineLearningMethod
:MachineLearningMethod rdf:type owl:Class ;
                       rdfs:subClassOf :Method ;
                       rdfs:comment "A method employing machine learning techniques."@en ;
                       rdfs:label "Machine Learning Method"@en .

### http://example.org/ai-ontology#MachineLearningModel
:MachineLearningModel rdf:type owl:Class ;
                      owl:equivalentClass [ rdf:type owl:Restriction ;
                                              owl:onProperty :usesTrainingAlgorithm ;
                                              owl:someValuesFrom :TrainingAlgorithm
                                              ] ;
                      rdfs:subClassOf :Model ;
                      rdfs:comment "ML model trained by algorithms."@en ;
                      rdfs:label "Machine Learning Model"@en ;
                      skos:exactMatch <https://cso.kmi.open.ac.uk/topics/machine_learning> .

### http://example.org/ai-ontology#Method
:Method rdf:type owl:Class ;
        rdfs:comment "AI method used."@en ;
        rdfs:label "Method"@en .

```

```

### http://example.org/ai-ontology#Model
:Model rdf:type owl:Class ;
      rdfs:comment "Generic AI model."@en ;
      rdfs:label "Model"@en .

### http://example.org/ai-ontology#NaturalLanguageProcessing
:NaturalLanguageProcessing rdf:type owl:Class ;
                           rdfs:subClassOf :ApplicationArea .

### http://example.org/ai-ontology#Paper
:Paper rdf:type owl:Class ;
       rdfs:subClassOf [ rdf:type owl:Restriction ;
                        owl:onProperty :hasAuthor ;
                        owl:minQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                        owl:onClass :Author
                      ] ;
       rdfs:comment "Research paper."@en ;
       rdfs:label "Paper"@en ;
       skos:narrowMatch <http://purl.org/ontology/bibo/Document> ;
       skos:exactMatch <http://purl.org/ontology/bibo/AcademicArticle> .

### http://example.org/ai-ontology#Repository
:Repository rdf:type owl:Class ;
            rdfs:comment "Repository hosting resources."@en ;
            rdfs:label "Repository"@en .

### http://example.org/ai-ontology#RuleBasedMethod
:RuleBasedMethod rdf:type owl:Class ;
                 rdfs:subClassOf :Method ;
                 rdfs:comment "A method based on explicit rules."@en ;
                 rdfs:label "Rule-Based Method"@en .

### http://example.org/ai-ontology#RuleBasedModel
:RuleBasedModel rdf:type owl:Class ;
                rdfs:subClassOf :Model ;
                rdfs:comment "AI model based on explicit rules."@en ;
                rdfs:label "Rule-Based Model"@en ;
                skos:narrowMatch <https://cso.kmi.open.ac.uk/topics/expert_knowledge>

### http://example.org/ai-ontology#SymbolicMethod

```

```

:SymbolicMethod rdf:type owl:Class ;
    rdfs:subClassOf :RuleBasedMethod ;
    rdfs:comment "A method using symbolic approaches."@en ;
    rdfs:label "Symbolic Method"@en .

### http://example.org/ai-ontology#SymbolicModel
:SymbolicModel rdf:type owl:Class ;
    rdfs:subClassOf :RuleBasedModel ;
    rdfs:comment "Symbolic AI models."@en ;
    rdfs:label "Symbolic Model"@en ;
    skos:narrowMatch <https://cso.kmi.open.ac.uk/topics/knowledge_representation> .

### http://example.org/ai-ontology#Task
:Task rdf:type owl:Class ;
    rdfs:comment "AI task addressed."@en ;
    rdfs:label "Task"@en .

### http://example.org/ai-ontology#TraditionalMLMethod
:TraditionalMLMethod rdf:type owl:Class ;
    rdfs:subClassOf :MachineLearningMethod ;
    rdfs:comment "A method based on traditional machine learning (e.g. linear regression)."@en ;
    rdfs:label "Traditional ML Method"@en ;
    skos:narrowMatch <https://cso.kmi.open.ac.uk/topics/machine_learning> .

### http://example.org/ai-ontology#TraditionalMLModel
:TraditionalMLModel rdf:type owl:Class ;
    rdfs:subClassOf :MachineLearningModel ;
    rdfs:comment "Traditional ML algorithms."@en ;
    rdfs:label "Traditional ML Model"@en ;
    skos:narrowMatch <https://cso.kmi.open.ac.uk/topics/machine_learning> .

### http://example.org/ai-ontology#TrainingAlgorithm
:TrainingAlgorithm rdf:type owl:Class ;
    rdfs:comment "Algorithm for training models."@en ;
    rdfs:label "Training Algorithm"@en .

#####
#    Individuals
#####

### http://example.org/ai-ontology#ComputerVision
:ComputerVision rdf:type owl:NamedIndividual .

```

```
### http://example.org/ai-ontology#ComputerVisionClass
:ComputerVisionClass rdf:type owl:NamedIndividual ,
                        :ApplicationArea ;
                        rdfs:label "Computer Vision"@en .
```

```
### http://example.org/ai-ontology#NaturalLanguageProcessing
:NaturalLanguageProcessing rdf:type owl:NamedIndividual .
```

```
### http://example.org/ai-ontology#NaturalLanguageProcessingClass
:NaturalLanguageProcessingClass rdf:type owl:NamedIndividual ,
                                    :ApplicationArea ;
                                    rdfs:label "Natural Language Processing"@en .
```

```
#####
# Annotations
#####
```

```
:ComputerVision rdfs:comment "Subset di ApplicationArea per Computer Vision."@en ;
                 rdfs:label "Computer Vision"@en .
```

```
:NaturalLanguageProcessing rdfs:comment "Subset di ApplicationArea per NLP."@en ;
                            rdfs:label "Natural Language Processing"@en .
```

```
<https://cso.kmi.open.ac.uk/topics/expert_knowledge> rdfs:comment "Computer Science Ontology: Expert Knowledge (CSO)"@en ;
                                                        rdfs:label "Expert Knowledge (CSO)"@en .
```

```
<https://cso.kmi.open.ac.uk/topics/knowledge_representation> rdfs:comment "Computer Science Ontology: Knowledge Representation (CSO)"@en ;
                                                                rdfs:label "Knowledge Representation (CSO)"@en .
```

```
<https://cso.kmi.open.ac.uk/topics/machine_learning> rdfs:comment "Computer Science Ontology: Machine Learning (CSO)"@en ;
                                                        rdfs:label "Machine Learning (CSO)"@en .
```

```
<https://cso.kmi.open.ac.uk/topics/neural_networks> rdfs:comment "Computer Science Ontology: Neural Networks (CSO)"@en ;
                                                        rdfs:label "Neural Networks (CSO)"@en .
```

```
<https://cso.kmi.open.ac.uk/topics/neurosymbolic_ai> rdfs:comment "Computer Science Ontology: Neuro-Symbolic AI (CSO)"@en ;
                                                        rdfs:label "Neuro-Symbolic AI (CSO)"@en .
```

```
#####
#    General axioms
#####
```

```
[ rdf:type owl:AllDisjointClasses ;
  owl:members ( :Author
                  :Conference
                  :Dataset
                  :Evaluation
                  :Journal
                  :Method
                  :Model
                  :Paper
                  :Repository
                  :Task
                )
] .
```

```
[ rdf:type owl:AllDisjointClasses ;
  owl:members ( :DeepLearningMethod
                  :HybridMethod
                  :SymbolicMethod
                  :TraditionalMLMethod
                )
] .
```

```
[ rdf:type owl:AllDisjointClasses ;
  owl:members ( :DeepLearningModel
                  :HybridModel
                  :SymbolicModel
                  :TraditionalMLModel
                )
] .
```

```
### Generated by the OWL API (version 4.5.29.2024-05-13T12:11:03Z) https://github.com
```

Flusso di Interazione con l'Utente

1. Descrizione generale

L'interfaccia si rivolge a due tipologie di utenti:

- **Ricercatore/Studente:** vuole cercare paper, modelli e task all'interno dell'ontologia.

- **Amministratore/Curatore:** ha il permesso di aggiungere nuovi elementi (autori, paper, modelli, task) e di modificarne i metadati.

I principali *use case* sono:

1. **Ricerca paper per autore o task:** l'utente inserisce il nome di un autore o un titolo di task e ottiene in risposta l'elenco dei paper corrispondenti.
2. **Navigazione gerarchica:** l'utente esplora le aree applicative (Computer Vision, NLP, ecc.) per visualizzare i task e i modelli corrispondenti.
3. **Inserimento/modifica dati (solo per Amministratore):** aggiungere un nuovo paper con autore, data, task e modello associato.

Schema di Interfaccia (Mockup senza Grafica)

1. Schermata di Login (solo Amministratore)

```

+-----+
|                                     |
|               AI Systems Ontology  |
|-----+
| [ Logo Università ]                |
|                                     |
|   +-----+   +-----+           |
|   | Username: |   | (campo di input) | |
|   +-----+   +-----+           |
|                                     |
|   +-----+   +-----+           |
|   | Password: |   | (campo di input) | |
|   +-----+   +-----+           |
|                                     |
| [ Login button ]   [ Annulla ]     |
+-----+

```

2. Menu Principale (Utente Normale)

```

+-----+
|                                     |
|               AI Systems Ontology  |
|-----+
| Benvenuto, Ricercatore!           |
|                                     |
| [1] Ricerca Paper per Autore       |
| [2] Ricerca Paper per Task         |
| [3] Esplora Aree Applicative       |
| [4] Logout                         |
|                                     |
| Seleziona un'opzione [1-4]: [   ] |
+-----+

```

3. Interfaccia di Ricerca (Per Autore o Task)

Ricerca Paper	
Digita Nome Autore: [_____]	
Oppure	»
Seleziona Task: [Dropdown che mostra tutti i Task]	
[Cerca]	[Indietro]

4. Risultati della Ricerca

Risultati della Ricerca	
Elenco Paper trovati:	
1. "Efficient ResNet50 for Image Classification"	
Autori: Alice Smith, Bob Rossi	
Data: 2023-06-20 Venue: CVPR2023	
2. "YOLOv5: Real-Time Object Detection"	
Autori: Bob Rossi, Carla Bianchi	
Data: 2023-06-21 Venue: CVPR2023	
3. "Text Generation with Transformer Models"	
Autori: Giovanni Esposito	
Data: 2023-05-20 Venue: NeurIPS2023	
...	
Seleziona numero paper per vedere dettagli []	
[Indietro]	

5. Visualizzazione Dettagli Paper

Dettagli Paper Selezionato	
Titolo: "Efficient ResNet50 for Image Classification"	
Autori: Alice Smith, Bob Rossi	
Data: 2023-06-20	
Venue: CVPR2023	
Task: Image Classification	

```

| Modello: ResNet50 |
| Dataset: ImageNet |
| Metriche: Accuracy, Precision, Recall |
| |
| [ Indietro ai Risultati ] |
| [ Nuova Ricerca ] |
+-----+

```

Esempio di Interazione con Dati Reali

Scenario: Ricerca Paper per Autore

1. Schermata iniziale (Utente Normale):

```

+-----+
|                               |
|               AI Systems Ontology               |
|-----|
| Benvenuto, Ricercatore! |
| |
| [1] Ricerca Paper per Autore |
| [2] Ricerca Paper per Task |
| [3] Esplora Aree Applicative |
| [4] Logout |
| |
| Seleziona un'opzione [1-4]: [ 1 ] |
+-----+

```

L'utente digita 1 e preme Invio.

2. Interfaccia di Ricerca (Per Autore):

```

+-----+
|                               |
|               Ricerca Paper               |
|-----|
| Digita Nome Autore: [ Giovanni Esposito ] |
| |
| [ Cerca ] [ Indietro ] |
+-----+

```

L'utente inserisce **Giovanni Esposito** e seleziona **Cerca**.

(Supponendo che nella base dati esista un solo autore "Giovanni Esposito" e i seguenti risultati:)

- **Paper003:** "Text Generation with Transformer Models" – 2023-05-20 – Venue: NeurIPS2023

- **Paper010:** “New Frontiers in Text Generation” – 2024-02-15 – Venue: NeurIPS2023

3. Risultati della Ricerca:

```

+-----+
|                               |
|          Risultati della Ricerca          |
|-----|
| Elenco Paper trovati per autore "Giovanni Esposito": |
| |
| 1. "New Frontiers in Text Generation" |
|   Data: 2024-02-15   Venue: NeurIPS2023 |
| |
| 2. "Text Generation with Transformer Models" |
|   Data: 2023-05-20   Venue: NeurIPS2023 |
| |
| Seleziona numero paper per vedere dettagli [ 1-2 ] |
| [ Indietro ] |
+-----+

```

L'utente digita 1 per vedere i dettagli di “New Frontiers in Text Generation”.

4. Visualizzazione Dettagli Paper:

```

+-----+
|                               |
|          Dettagli Paper Selezionato          |
|-----|
| Titolo: "New Frontiers in Text Generation" |
| Autori: Giovanni Esposito, Anna Ferretti |
| Data: 2024-02-15 |
| Venue: NeurIPS2023 |
| Task: Text Generation |
| Modello: Transformer Model |
| Dataset: WikiText-103 |
| Metriche: BLEU, ROUGE |
| |
| [ Indietro ai Risultati ] |
| [ Nuova Ricerca ] |
+-----+

```

A questo punto l'utente può scegliere di tornare indietro per consultare altri paper, oppure avviare una nuova ricerca.

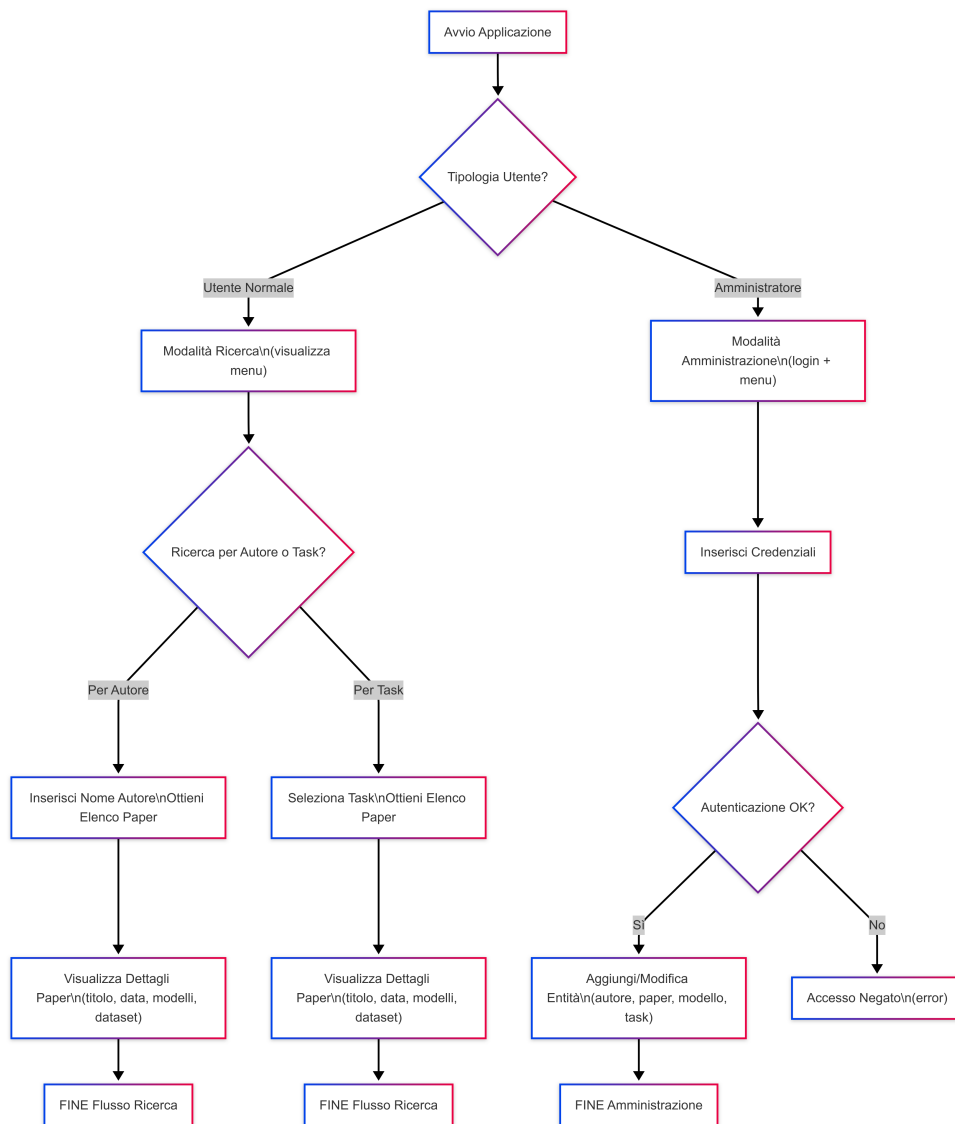


Figure 4: Flowchart che illustra l'ipotetico schema di interazione con l'utente

Note Conclusive

- Il **flow chart** mostra chiaramente la separazione tra utenti “Normali” e “Amministratori” e i percorsi logici di ricerca vs. gestione dati.
- Lo **schema di interfaccia** illustra con semplicità i mockup testuali, sufficienti per capire la disposizione dei campi e dei comandi senza ricorrere a elementi grafici.
- L'**esempio di interazione** con dati reali (autore “Giovanni Esposito”) dimostra come il sistema invii query SPARQL, riceva risultati e li visualizzi in modo intuitivo.

Note Conclusive

- Il **flow chart** mostra chiaramente la separazione tra utenti “Normali” e “Amministratori” e i percorsi logici di ricerca vs. gestione dati.

- Lo **schema di interfaccia** illustra con semplicità i mockup testuali, sufficienti per capire la disposizione dei campi e dei comandi senza ricorrere a elementi grafici.
- L'**esempio di interazione** con dati reali (autore “Giovanni Esposito”) dimostra come il sistema invii query SPARQL, riceva risultati e li visualizzi in modo intuitivo.

Sitografia

- Ontotext. “The InnoGraph Artificial Intelligence Taxonomy.” <https://www.ontotext.com/blog/the-innograph-artificial-intelligence-taxonomy/>
- Computer Science Ontology (CSO). <https://cso.kmi.open.ac.uk/home>
- Wikipedia, “Artificial intelligence” (edizione inglese). https://en.wikipedia.org/wiki/Artificial_intelligence
- Wikipedia, “Research” (edizione inglese). <https://en.wikipedia.org/wiki/Research>