

Engineering an ontology for Age-Related Macular Degeneration using Agentic AI and DL-learner

Calina Annemary Borzan¹, Alexandru Lecu¹, George Adrian Muntean²,
Simona Delia Nicoara² and Adrian Groza¹

¹Artificial Intelligence Research Institute AIRi@UTCN, Technical University of Cluj-Napoca, 400114 Cluj-Napoca, Romania

²Department of Ophthalmology, "Iuliu Hatieganu" University of Medicine and Pharmacy, Emergency County Hospital, 400347 Cluj-Napoca, Romania

Abstract

Biomedical ontologies offer a principled solution, formally representing diseases, treatments, and genes as semantically linked entities. Yet no continuously updatable ontology exists for AMD specifically. This paper presents a fully agentic pipeline using three specialised LangChain agents: for literature discovery, knowledge extraction, and quality assurance, combined with DL-Learner for axiom discovery. The result is an OWL ontology that remains current with AMD literature through incremental monthly updates.

Keywords

Age-Related Macular Degeneration, Biomedical Ontology, Agentic AI, LangChain, OWL, DL-Learner

1. Introduction

Ontologies provide a formal, machine-readable representation of domain knowledge that supports reasoning and semantic search. Biomedical ontologies are used to standardize heterogeneous medical data and to support clinical knowledge discovery through OWL 2 reasoning, SPARQL querying and description-logic class expressions [1]. We aim here to develop an ontology for Age-Related Macular Degeneration (AMD).

AMD is one of the leading causes of visual impairment and blindness, especially among the elderly population. AMD is particularly complex, arising from a combination of genetic predisposition, environmental exposure, and lifestyle factors such as smoking and cardiovascular disease. However, building such an ontology for AMD from the literature is not straightforward. AMD abstracts are dense with specialised terminology, abbreviations, and trade names: terms such as *anti-VEGF*, *PDT*, or *AREDS* are routine in clinical texts but require domain background to interpret correctly. Relations between entities are frequently implicit rather than stated directly, an abstract may describe a drug's mechanism of action without ever using the word *treats*, which complicates automated extraction. Beyond extraction, keeping the ontology current is a practical challenge: new AMD clinical trials appear on PubMed each month, so an ontology built once quickly becomes incomplete.

In a preliminary experiment applying the SCHEMA-MINERpro framework [2] to 100 AMD abstracts, we encountered several practical problems with passing the full schema to the model at each refinement step (Table 1). These problems, together with the need to keep the ontology current, as new AMD abstracts are published each month, led us to build a multi-agent system: one agent retrieves new abstracts from PubMed, a second processes them and proposes additions to the ontology through validated tool calls, and a third validates the assembled ontology and proposes corrections to a human reviewer.

The contributions of this paper are: (i) a rule-validated extraction pipeline in which every proposed class, instance, and triple is checked against a closed predicate vocabulary, an OWL 2 disjointness schema, and a source-grounding rule before being committed; (ii) DL-Learner CELOE experiments recovering non-trivial OWL 2 class expressions with 91.3% average accuracy across 17 learning problems

RuleML+RR'26: Companion Proceedings of the 10th International Joint Conference on Rules and Reasoning, August 24–26, 2026, Vilnius, Lithuania

✉ borzan.fl.calina@student.utcluj.ro (C. A. Borzan)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

and 77–81% UMLS alignment; (iii) a human-in-the-loop interface that allows a clinical expert to review, accept, or reject individual corrections.

Section 2 reviews related work on LLM-based ontology engineering and AMD knowledge graphs. Section 3 analyses the failures that motivated our redesign. Section 4 describes the system architecture. Section 5 presents the running scenario and evaluation.

2. Related work

The integration of large language models has transformed automated ontology engineering, replacing manual translation of domain knowledge into formal axioms that historically required months of expert work. Mateiu and Groza [3] developed a Protégé plugin that uses a fine-tuned GPT-3 to translate natural language descriptions into OWL Functional Syntax, training the model on 150 prompt-response pairs across zero-shot, few-shot and fine-tuning strategies, with fine-tuning proving most effective. NeOn-GPT [4] extends this idea to cross-domain ontology learning, combining proprietary (GPT-4o) and open-source (Mistral, Llama-4, DeepSeek) models to bootstrap ontologies across four domains following the NeOn methodology, with explicit reuse of existing domain ontologies to guide modeling. OntoGenix [5] proposes an LLM-assisted framework for ontology engineering directly from structured datasets, with iterative refinement against the input schema. Across these systems, generated axioms must still be manually validated, which limits scalability when processing thousands of medical abstracts; our work addresses this by integrating DL-Learner for automated validation and axiom discovery directly on the assembled knowledge graph. Approaches based on declarative shape constraints, such as SHACL [6] provide an alternative for KG validation, but require manually authored shapes; our pipeline derives the validation rules from the upper ontology itself.

DL-Learner [7] provides machine-learning support for ontology engineering by searching the space of OWL 2 class expressions for a description that separates positive and negative examples provided by the user. Its CELOE algorithm (Class Expression Learning for Ontology Engineering) is guided by refinement operators over OWL 2 DL and remains the de facto baseline for class expression learning over knowledge graphs. Fanizzi and d’Amato have extended this line of work to probabilistic axiom induction, learning interpretable Naive Bayes classifiers over DL knowledge bases, and converting them into probabilistic axioms [8]. We use CELOE for axiom discovery (Section 5), defining 17 learning problems over the extracted AMD knowledge graph that recover non-trivial OWL 2 class expressions across property restrictions, intersections, and cardinality patterns.

For AMD specifically, Lecu, Groza, and Hawizy [9] created CausalAMD, an AMD-specific knowledge graph with automated extraction. They fine-tuned GPT-3.5 Turbo for Named Entity Recognition (12 entity types) and Relation Extraction (8 causal relations), reaching 76.9% precision on 226 extracted relations, with the resulting ontology augmented through HerMiT [10] reasoning and queried through a LangChain-based RAG interface over Ontotext GraphDB. Subsequent work [11] scaled the approach by replacing fine-tuning with few-shot prompting of GPT-4o-mini, extracting over 43,000 causal relations from 5,000 AMD abstracts, indexing them in a Weaviate vector database, and evaluating eight language models on 3,000 queries to show that 3B-parameter models with RAG can match or exceed 70B-parameter baselines. In the broader context of ophthalmology, Gilson et al. [12] constructed an ontology covering 1,497 EyeWiki articles using SciBERT-based NER, identifying 7,600 unique concepts, and demonstrated that retrieval-augmented generation reduced hallucinated citations from 48% to 16% on known questions. We build on the methodological precedent of both systems while moving from one-shot extraction to a continuously updatable pipeline with rule-validated proposals.

Beyond single-LLM systems, recent work has organized LLM-based extraction as cooperating agents. The Agentic Medical Graph-RAG framework [13] uses LLM-driven agents with domain-specific search tools to build and refine medical knowledge graphs for question answering, reaching a 74.1% F1-score on MEDQA. SCHEMA-MINERpro [2] uses LLM agents to align the extracted properties with the QUDT ontology through iterative semantic search. Both demonstrate that agent decomposition mitigates token limits and enables fine-grained extraction, but both rely on the LLM to reproduce or reason over

Table 1
SCHEMA-MINERpro with Llama 3.1:8B across three stages.

Stage	Classes	Properties	Instances	Triples
Stage 1	25	10	107	10
Stage 2	40	22	126	18
Stage 3	50	45	56	0

the full schema at each step, with no declarative constraints on what proposals are admissible. Our pipeline differs in that the LLM never sees or rewrites the schema: it submits structured deltas against a compact summary, and every proposal is checked against a closed predicate vocabulary, an OWL 2 disjointness schema, and a source-grounding rule before being committed (Section 4).

3. Motivation

3.1. Applying SCHEMA-MINERpro on AMD

Before building our pipeline, we evaluated SCHEMA-MINERpro [2] on the 100 clinical-trial abstracts of AMD collected by Lecu et al. [11]. The goal was to understand what an existing agentic schema extraction tool can and cannot do on AMD data and to use those findings to inform our own design.

SCHEMA-MINERpro uses a four-stage iterative process. Stage 1 generates a baseline schema from a domain specification. Stage 2 refines the schema over a chosen set of abstracts. Stage 3 validates and extends it against the full data set. Stage 4 optionally grounds properties to the QUDT ontology. Since the framework was designed for full-text articles, we adapted the prompts to work with abstracts and provided a manually curated AMD domain specification as input. Two changes were necessary: the document-level chunking instruction was replaced with a note that the input is a 150–300 word abstract, and a list of canonical AMD terms (*DryAMD*, *WetAMD*, *CFH*, *ARMS2*) was added to the domain specification to reduce naming inconsistencies across abstracts.

Results with Llama 3.1:8B. Table 1 shows how the schema evolved across stages. Stage 1 produced a reasonable starting point: 25 classes, 10 reusable predicates, 107 instances, and 10 grounded triples such as *Ranibizumab treats WetAMD*. Stage 2 refined the structure over 20 curated abstracts, growing to 126 instances and 18 triples. However, Stage 3 exhibited a clear regression. Despite processing 100 abstracts, it dropped to zero example triples and reorganised the 56 remaining individuals into a dictionary without class membership. Object properties retained only declarative domain and range metadata, losing all grounded examples. The root cause was the growing schema JSON exceeding the model’s effective context, leaving almost no capacity to process the current abstract.

Results with Llama 3.3:70B. The 70B run exhibited a different but equally serious failure. With more generation capacity, the model produced 228 classes of which 91% had no instances and no subclasses. These were not meaningful ontological categories but decorative classes lifted from individual abstract mentions, for example *IllDefinedChoroidalNeovascularization*, *NightTimeLightTherapy*, each appearing once in the literature with no justification for being a standalone class. The predicate vocabulary grew to 52 properties. Many were instance-specific variants of the same logical relation: *improvesBCVA*, *improvesReadingSpeed*, instead of a single reusable *improves* predicate. Most critically, the 70B run produced zero instances and zero triples across all stages, as shown in Table 2: it knew *treats* exists as a relation but never extracted *Ranibizumab treats WetAMD*.

3.2. Limitations That Motivated the Redesign

Beyond the context overflow already observed in the 8B run, two further problems emerged consistently across both configurations. Because the model had to reproduce all entity names at each stage,

Table 2

SCHEMA-MINERpro with Llama 3.3:70B across three stages.

Stage	Classes	Properties	Instances	Triples
Stage 1	35	3	0	0
Stage 2	99	3	0	0
Stage 3	228	52	0	0

Table 3

Failure modes and corresponding design choices.

Failure	Design choice
Schema JSON exceeds context window	Agent sees only a compact ontology summary, never the full schema
Spelling variants accumulate as duplicates	Fuzzy deduplication (Levenshtein ≤ 2) in the delta validator
Class explosion: one class per mention	Pending-class queue: a class is added only once two instances support it
Predicate proliferation	Closed predicate vocabulary; no new predicates during extraction
Zero grounded triples in Stage 3	One structured tool call per abstract; the triple step cannot be skipped

small spelling differences such as “ChoroidalNeovascularization” and “ChoroidalNeovascularisation” accumulated as separate entities with no mechanism to detect or merge them. Across both runs, early stages produced far more classes than instances (up to 140 classes with only 54 instances), as the model created a new class for each entity rather than adding it as an instance. Table 3 maps each failure mode to the design decision that addresses it in our pipeline.

Core design principle: *the model sees only a compact summary of the current ontology and proposes additions one abstract at a time through a single validated tool call, never reading or rewriting the full schema.*

4. System Architecture

Figure 1 shows the proposed solution: an upper ontology is initialised with seven root classes; a literature discovery agent retrieves abstracts and feeds them to an extraction agent; a validation agent audits the result for human review; the ontology is then exported to OWL, checked with HermiT, and analysed with DL-Learner.

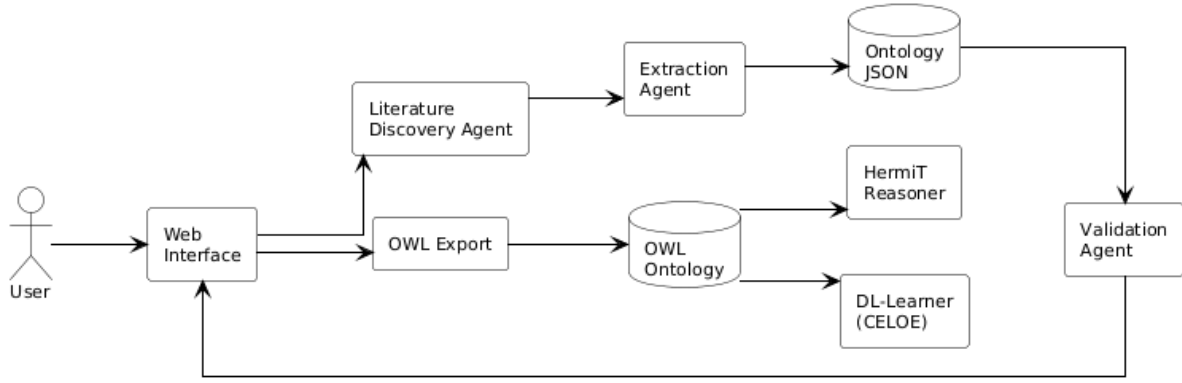


Figure 1: End-to-end pipeline: literature discovery, extraction, validation, Hermit reasoning, and DL-Learner axiom discovery.

4.1. Schema Rules

The pipeline starts from a small hand-crafted ontology rather than importing an existing biomedical resource. We drew inspiration from established biomedical ontologies such as MONDO [14], DOID [15], and the OBO Foundry [16] in defining the class structure. However, these resources represent AMD as a single disease entry with no treatment or biomarker subhierarchy, and importing them in full would introduce concepts outside our scope. We therefore designed a compact domain-specific schema from scratch, retaining only the relevant terms for our system. Seven root classes were selected: *Disease*, *Treatment*, *Biomarker*, *DiagnosticMethod*, *RiskFactor*, *ClinicalOutcome*, and *MolecularTarget*. Nine object properties form the closed predicate vocabulary: *treats*, *inhibits*, *causesOrIncreases*, *indicates*, *diagnosedBy*, *hasSymptom*, *measuredBy*, *associatedWith*, and *assessedBy*.

The vocabulary is closed, no new predicates can be introduced during extraction, to prevent the predicate explosion observed in the pre-agentic runs (Section 3). No instance of a clinically meaningful relation being dropped due to the closed vocabulary was observed in this corpus; however, this remains a risk as the vocabulary is extended to larger, more heterogeneous corpora, and is a tradeoff we accept in exchange for eliminating predicate proliferation. To prevent the model from assigning the same entity to semantically incompatible classes, we declared a disjointness constraint between *Disease*, *Treatment*, *DiagnosticMethod*, and *ClinicalOutcome*.

4.2. Dataset

To build the ontology, we used 100 AMD clinical-trial abstracts from the LLMKGraph corpus [11], collected from PubMed and restricted to English-language clinical trials and observational studies. Of these, 20 abstracts were manually selected for Stage 2 based on topical diversity: we chose abstracts covering distinct AMD subtypes, treatment modalities, and biomarker categories so that the class structure refined in Stage 2 would generalise to the full corpus in Stage 3. All 100 abstracts are processed in Stage 3 to populate instances and triples.

The choice of 20 abstracts for Stage 2 and 100 for Stage 3 reflects a deliberate progressive loading strategy. Stage 2 uses a small, manually curated set chosen to cover as many distinct aspects of AMD as possible, ensuring the core schema is populated with representative entities before the noisier full corpus is introduced in Stage 3. Processing the full corpus directly, without this intermediate step, was found to produce more hallucinations and class explosions in preliminary testing.

All three agents are implemented using LangChain, a framework that simplifies the integration of large language models with external tools and structured workflows and specific prompts making it well suited for building modular, inspectable pipelines where each step can be validated independently.

4.3. Pipeline Agents

Literature discovery agent. The literature discovery agent uses a ReAct reasoning loop to autonomously compose and refine PubMed queries until it has identified a relevant set of new abstracts. It queries PubMed via the NCBI Entrez API [17] using AMD-specific search terms, retrieving abstracts published within a configurable time window specified by the user. Retrieved abstracts are deduplicated by PubMed identifier and passed to the extraction agent.

Extraction agent. The extraction agent uses LangChain’s tool-calling pattern: the language model receives a compact summary of the current ontology and must submit exactly one structured proposal per abstract, which the system validates before applying. Two models are supported: Llama-3.3-70B via the Groq API (default, used in the web interface) and Qwen2.5:32B running locally via Ollama (used when no internet connection or API key is available). The agent runs in three stages, each driven by a different prompt template. Stage 1 processes a domain specification written by hand listing core AMD terminology (*DryAMD*, *WetAMD*, *CFH*, *Ranibizumab* etc.), and uses it to initialise the class hierarchy. Stage 2 refines the structure over 20 curated abstracts. Stage 3 extracts instances and triples from all 100 abstracts. After each stage, the pipeline pauses and presents the current ontology summary to the domain expert, who can review the state before proceeding. Each abstract’s contribution is structured as a delta of three operations: *ADDCLASS*, *ADDINSTANCE*, and *ADDTUPLE*, following the Knowledge Graph Change Language standard [18]. Before any item is committed, Algorithm 1 applies a sequence of checks across three groups (one for classes, four for instances, three for triples) to catch the failure modes documented in Section 3. A proposed class is held in a pending queue until at least two instances support it, preventing under-populated classes. Each instance must appear in the source abstract [19] to prevent hallucination, must not be a personal name, must not duplicate an existing entity, and must not be assigned to a class disjoint from one it already inhabits: for example, the same entity cannot be both a Disease and a Treatment. Each triple must use a predicate from the closed registry, reference entities already in the ontology, and not duplicate an existing triple. Of the final ontology, the 11 classes form a fixed skeleton seeded by hand; all 104 instances and 114 triples were learned entirely from the corpus. Across the 100-abstract run, the failure modes encountered in practice matched the five categories in Table 3; no additional failure pattern requiring a new validation rule was observed.

Prompt engineering. Three design choices proved relevant. First, explicitly stating that the input is a short research abstract prevented the model from discarding valid entities for lacking dosage or follow-up data. Second, providing a list of canonical AMD terms (*CFH*, *ARMS2*) reduced naming inconsistencies across abstracts. Third, including relation examples such as *Ranibizumab treats WetAMD* directly in the prompt reduced invalid predicates to zero across the final corpus.

Validation agent. The extraction agent processes one abstract at a time and can only catch local errors. Broader mistakes (the same concept spelled in two different ways, a triple in the wrong direction, or a drug filed under *Biomarker* instead of *Treatment*) only appear once the full ontology is assembled. The validation agent addresses this using LangChain’s ReAct pattern, which lets the model decide what to inspect next based on what it just found.

The agent has twelve tools: four for read-only inspection, six deterministic scanners (OWL 2 punning, dual parents, self-referential triples, spelling duplicates within Levenshtein distance 2, biomarker-semantics heuristics, and domain/range violations), one external MeSH lookup [20], and one tool that records a correction for human review. Eleven tools are pure Python; the model reads their output, applies medical judgement, and selects among five correction types: *remove*, *swap*, *move*, *change predicate*, and *swap and change predicate*. The agent runs up to five passes; no correction is applied without user approval.

One limitation concerns error propagation across agents. An entity incorrectly extracted by the extraction agent that is not caught during the validation agent’s structural checks will persist in the final ontology. The current pipeline has no mechanism for tracking the provenance of individual triples back to their source abstract, which makes it difficult to identify and correct systematic extraction errors after the fact. The validation agent reduces this risk through deterministic structural scanners and human review at every stage, but does not eliminate it entirely, since it operates heuristically rather

Algorithm 1 Delta validation for a single abstract

Require: Abstract a , ontology $\mathcal{O}=(C, I, T, P)$, summary $\sigma(\mathcal{O})$, pending queue Q

Ensure: Updated $\mathcal{O}'$, rejection log R , updated Q

```
1:  $\Delta \leftarrow \text{LLM-AGENT}(a, \sigma(\mathcal{O}))$ ;  $R \leftarrow \emptyset$ 
2: for all class  $c \in \Delta_C$  do
3:   if  $|\text{children}(c)| < 2$  then  $Q \leftarrow Q \cup \{c\}$  ▷ defer until 2+ instances seen
4:   else  $C \leftarrow C \cup \{c\}$ 
5:   end if
6: end for
7: for all instance  $i \in \Delta_I$  do
8:   if  $\text{NORM}(i) \notin \text{NORM}(a)$  then  $R \leftarrow R \cup \{i\}$ ; continue ▷ not in source abstract
9:   end if
10:  if  $i$  starts with personal title (Dr., Mr., etc.) then  $R \leftarrow R \cup \{i\}$ ; continue ▷ personal name
11:  end if
12:  if  $\text{NORM}(i) \in \text{NORM}(I \cup C)$  then  $R \leftarrow R \cup \{i\}$ ; continue ▷ duplicate
13:  end if
14:  if  $\text{root}(i.\text{class}) \perp \text{root}_{\text{cur}}(i)$  then  $R \leftarrow R \cup \{i\}$ ; continue ▷ disjoint class conflict
15:  end if
16:   $I \leftarrow I \cup \{i\}$ 
17: end for
18: for all triple  $(s, p, o) \in \Delta_T$  do
19:   if  $p \notin P$  or  $\{s, o\} \not\subseteq I \cup C$  or  $(s, p, o) \in T$  then  $R \leftarrow R \cup \{(s, p, o)\}$ ; continue
20:   end if
21:    $T \leftarrow T \cup \{(s, p, o)\}$ 
22: end for
23: return  $\mathcal{O}'$ ,  $R$ ,  $Q$ 
```

than through formal verification.

OWL export. Once the validation cycle completes, the ontology is serialised to OWL/XML, emitting class declarations, `rdfs:subClassOf` edges, named individuals, object-property assertions, and pairwise `owl:disjointWith` axioms. The resulting file is the input to the evaluation in Section 5.

All pipeline stages are accessible through a browser-based interface built with Vue 3 and FastAPI. The interface provides real-time log streaming during extraction and validation, a graph-based ontology viewer with class and relation filters, and dedicated review panels where the human expert accepts or rejects individual extraction proposals and validation fixes without touching any code. HerMiT reasoning and DL-Learner experiments can be triggered and monitored from the same interface, and the ontology snapshot manager allows the reviewer to save and restore named checkpoints before committing a set of changes.

5. Running Scenario and System Evaluation

5.1. Running scenario

We trace the full pipeline on Example 5.1: extraction, human review, HerMiT reasoning, and evaluation against external biomedical resources. [Running abstract on AMD] “Aging contributes to the risk of development of ocular diseases including, but not limited to, Age-related Macular Degeneration (AMD) that is a leading cause of blindness in the United States as well as worldwide. Retinal aging, that contributes to AMD pathogenesis, is characterized by accumulation of drusen deposits, alteration in the composition of Bruch’s membrane and extracellular matrix, vascular inflammation and dysregulation, mitochondrial dysfunction, and accumulation of reactive oxygen species (ROS), and subsequent retinal pigment epithelium (RPE) cell senescence. Since there are limited options available for the prophylaxis and treatment of AMD,

new therapeutic interventions are constantly being looked into to identify new therapeutic targets for AMD. This review article discusses the potential candidates for AMD therapy and their known mechanisms of cytoprotection in AMD. These target therapeutic candidates include APE/REF-1, MRZ-99030, Ciliary NeuroTrophic Factor (CNTF), RAP1 GTPase, Celecoxib, and SS-31/Elamipretide.”

Starting ontology. The seed of Figure 2(a) contains four root classes (*Disease*, *Treatment*, *Biomarker*, *RiskFactor*), one subclass ($AMD \sqsubseteq Disease$, with *WetAMD* below it), three instances (*Celecoxib*, *Drusen*, *WetAMD*), four predicates (*treats*, *indicates*, *associatedWith*, *causesOrIncreases*), and one seed triple (*Drusen indicates AMD*). The *RiskFactor* class is intentionally empty so that enrichment is visible.

Extracting entities and relations. The extraction agent receives the 141-word abstract together with a compact summary of the current ontology and submits one structured proposal containing **nine instances**, **nine relations**, and **one candidate root class** (*MolecularTarget*, to house *RAP1GTPase*) that is not in the seed. Algorithm 1 auto-filters two items before the reviewer sees anything: *Drusen indicates AMD* is dropped as a duplicate of an existing seed triple, and *AMDcausesBlindness* is dropped because *Blindness* is not in the closed vocabulary. The remaining proposal is shown to the human reviewer in the web interface and summarised in Listing 1. To keep the running demo compact, the reviewer accepts only four instances and four relations and *defers the remaining five proposals, including the request for a new MolecularTarget class, for future iterations*. Figure 2a shows the seed before extraction; Figure 2b shows the ontology after the reviewer commits the accepted subset.

Listing 1: Extraction agent output for Example 5.1: nine instances and nine relations proposed, two auto-filtered, four instances and four relations accepted by the reviewer.

```

1  PROPOSED INSTANCES (9 across 4 classes -- 1 class new)
2  Biomarker : BruchsMembrane, ROS, RPE
3  RiskFactor : Aging
4  Treatment : APEREF1, CNTF, MRZ99030, SS31Elamipretide
5  MolecularTarget : RAP1GTPase [new class]
6  PROPOSED RELATIONS (9)
7  APEREF1 treats AMD
8  CNTF treats AMD
9  MRZ99030 treats AMD
10 SS31Elamipretide treats AMD
11 Celecoxib treats AMD (existing)
12 Aging causesOrIncreases AMD
13 BruchsMembrane associatedWith AMD
14 ROS associatedWith AMD
15 RPE associatedWith AMD
16 AUTO-FILTERED (2)
17 - Drusen indicates AMD [duplicate of an existing seed triple]
18 - AMD causesBlindness [object 'Blindness' not in closed vocabulary; out-of-schema guard]
19 REVIEWER ACCEPTED (4 instances + 4 relations)
20 Biomarker : BruchsMembrane
21 RiskFactor : Aging
22 Treatment : APEREF1, CNTF
23 Triples : BruchsMembrane associatedWith AMD
24 Aging causesOrIncreases AMD
25 APEREF1 treats AMD
26 CNTF treats AMD
27 REVIEWER DEFERRED (5 instances + 5 relations + 1 new class)
28 Biomarker : ROS, RPE
29 Treatment : MRZ99030, SS31Elamipretide
30 MolecularTarget : RAP1GTPase [new-class request not granted]
```

The three automatic filters in Listing 1 act before the reviewer sees anything, reducing manual effort. The four accepted relations all connect through the shared object *AMD*, illustrating how new contributions are anchored to the current ontology state rather than introduced as isolated triples.

Reasoning with HermiT. The exported OWL file is loaded into Protégé 5.6 and checked with HermiT 1.4. On the full ontology (11 classes, 104 instances, 114 asserted triples) the reasoner reports zero unsatisfiable classes and infers 38 additional axioms, primarily through the `owl:disjointWith` declarations between *Disease*, *Treatment*, *DiagnosticMethod*, and *ClinicalOutcome* and through domain/range constraints on the predicates. For the four triples committed from the running abstract, the reasoner derives

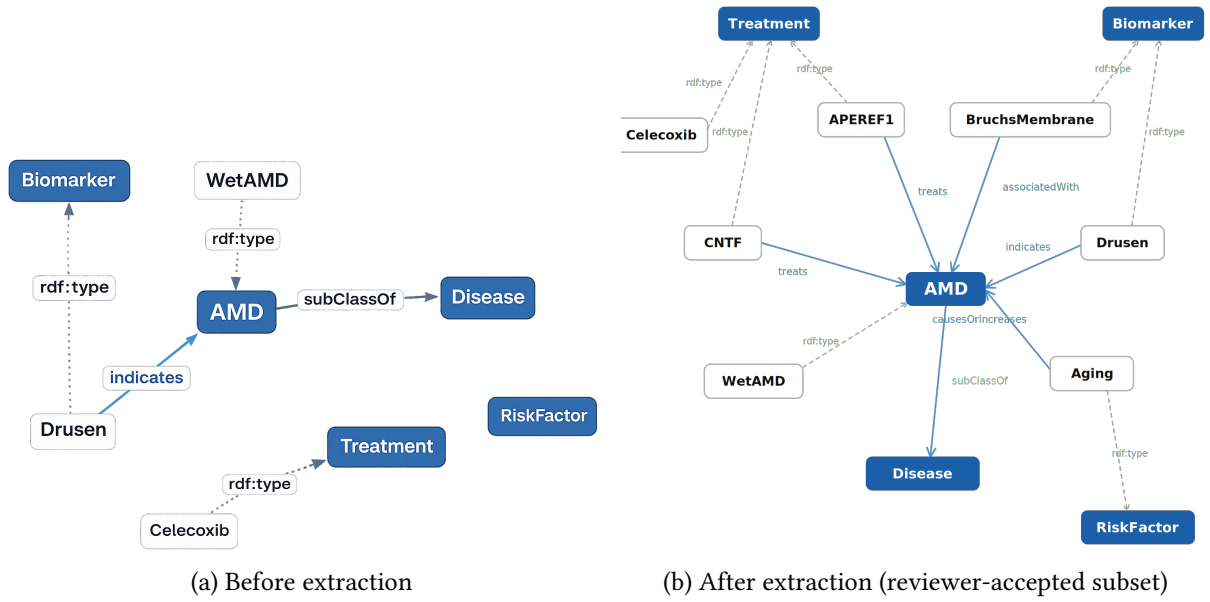


Figure 2: Ontology graph before and after the extraction agent processes Example 5.1

the type assertions in Listing 2 by applying the domain/range constraints and the *Disease/Treatment* disjointness axiom already present in the full ontology.

Listing 2: HermiT inferences for the four committed triples from Example 5.1, derived from property domain/range and disjointness axioms.

```

1 { AMD, rdf:type, Disease } (range of treats)
2 { APEREF1, rdf:type, Treatment } (domain of treats)
3 { CNTF, rdf:type, Treatment }
4 { Aging, rdf:type, RiskFactor } (domain of causesOrIncreases)
5 { BruchsMembrane, rdf:type, Biomarker } (domain of associatedWith)
6 { APEREF1, NOT rdf:type, Disease } (Disease/Treatment disjointWith)
7 { CNTF, NOT rdf:type, Disease }

```

5.2. Evaluation

We evaluate the pipeline along four axes: domain coverage on patient-relevant questions, alignment with biomedical terminology (UMLS and four external KGs), clinical correctness assessed by an ophthalmologist, and learned axioms via class-expression learning (DL-Learner).

Domain coverage via competency questions. We translate the 143 AMD patient questions collected by Muntean et al. [21] into SPARQL and execute them against the ontology. A question is counted as answered only if the query returns entities that cover the medical content asked about. As the right half of Table 4 shows, the ontology answers 28% of questions overall, with clinical categories reaching 31–38% and lifestyle questions trailing at 19%. The gap reflects topics absent from clinical-trial abstracts: diet, sunlight, and daily living draw on knowledge that abstracts rarely report.

Alignment with biomedical terminology. We ground each entity against UMLS using two independent methods (Semantic Types and NIH Semantic Groups) and check overlap with four external knowledge graphs (MeSH [20], MONDO [14], DOID [15], HPO [22]). Precision is matches divided by matches plus mismatches; coverage is the share of entities found in each external resource. As the left half of Table 4 shows, UMLS assigns 78–81% of extracted entities to the correct semantic type, and 85% appear in at least one external knowledge graph, both about 20 percentage points above the pre-agentic baseline. This confirms that the validation filters push extraction towards canonical medical terms.

The pre-agentic baseline uses the ontology produced by SCHEMA-MINERpro with Llama 3.1:8B on the same 100-abstract corpus (Section 3).

Table 4

Alignment with biomedical terminology.

Metric	Pre-agentic	Agentic
UMLS TUI precision	57.14%	77.66%
UMLS Sem. Groups	60.50%	80.73%
Coverage in MeSH	60%	82%
Coverage in MONDO	8%	28%
Coverage in DOID	1%	13%
Coverage in HPO	8%	12%
Any KG	63%	85%

Category	Total	Answered
General questions	16	6 (38%)
Changes in vision	38	11 (29%)
Treatment for wet AMD	29	9 (31%)
Support services	13	4 (31%)
Contacting professional	10	3 (30%)
Diet and lifestyle	37	7 (19%)
All	143	40 (28%)

It should be noted that the pre-agentic SCHEMA-MINERpro baseline differs from the proposed pipeline in both architecture and structural constraints simultaneously. The observed improvements therefore reflect the combined effect of the multi-agent design and the constraints introduced in this work, and cannot be attributed to either factor in isolation. A controlled ablation isolating each factor remains as future work.

Axiom discovery with DL-Learner. We tested whether the extracted ontology contains enough structure for an automated reasoner to recover non-trivial class definitions. DL-Learner 1.5 [7] takes a target class with positive and negative individuals and searches the OWL 2 class expression space for one that covers the positives and excludes the negatives.

We defined seventeen learning problems spanning Disease subtypes, drug mechanisms, biomarker discrimination, and diagnostic-method categorisation. Each problem was first run once with default settings (CELOE, predicted accuracy, 60 s, 0% noise) to obtain a baseline. **Six representative problems were then subjected to a full parameter sweep:** two algorithms (CELOE, OCEL), three accuracy metrics (predicted accuracy, F-measure, Jaccard distance), three time budgets (30/60/120 s), and three noise tolerances (0/10/25%): 45 configurations per problem (CELOE with two metrics $\times$ 3 time budgets $\times$ 3 noise levels = 18; OCEL with three metrics $\times$ 3 $\times$ 3 = 27), 270 sweep runs in total on top of the 17 baseline runs. **Table 5** reports the best CELOE result per problem; **Table 6** drills down on the six sweep problems: one selected from each major category (disease subtype, drug mechanism, biomarker typing, diagnostic separation, relation-based classification, and a boundary case expected to fail), to test whether algorithm and metric choice changes the outcome across structurally different learning problems.

The expressions recovered in Table 5 combine multiple OWL 2 DL constructs: simple class membership (RiskFactor, Treatment); property restrictions (inhibits some MolecularTarget); intersection of class and property restriction (Biomarker $\sqcap$ measuredBy some DiagnosticMethod); negation combined with cardinality ((not AntiVEGFTherapy) $\sqcap$ (treats max 1 Thing), captured by problem 7, encoding the clinical definition of non-anti-VEGF therapy that treats a single condition); union expressions (AntiVEGFTherapy $\sqcup$ (treats min 3 Thing), problem 3); and implicit associatedWith Disease patterns recovered on problem 17, distinguishing genetic biomarkers (CFH, ARMS2, HTRA1) from cellular and anatomical ones. Sixteen problems in Table 5 reach 80–100% accuracy. Problem 16 (imaging_vs_functional, 50%) is a corpus limitation: CELOE

Table 5

Best CELOE result across 17 learning problems. S=single run; W=full 45-config sweep. Bold marks 100% accuracy.

	Problem	T	Acc	F1
1	amd_genes	S	90.0	91.7
2	vegf_inhibitors	W	95.0	94.7
3	wetamd_drugs	S	94.7	94.1
4	amd_biomarkers	W	91.7	88.9
5	risk_factors	S	100	100
6	vegf_ELTL	S	90.0	90.0
7	dryamd_treatments	W	100	100
8	diagnosable_diseases	S	100	100
9	targeted_vs_general	W	88.0	87.0
10	dual_mechanism	S	100	100
11	classlearn_Treat	S	92.7	93.9
12	classlearn_Gene	S	96.2	96.0
13	posonly_antivegf	S	100	100
14	missing_inhibits	S	80.8	81.5
15	missing_assoc	S	82.6	84.6
16	imaging_vs_funct	W	50.0	66.7
17	genetic_vs_cellular	W	100	100
Average: 91.3% acc / 92.3% F1 6 problems at 100%, 1 struggle (problem 16)				

Table 6

Multi-metric sweep on six W-marked problems. Each cell shows best accuracy (%) across 9 runs (3 time budgets $\times$ 3 noise levels). *skip*=CELOE+Jaccard excluded [23]; *FAIL*=all 9 runs crash.

Problem	CELOE			OCEL		
	pred.acc	F-meas.	Jaccard	pred.acc	F-meas.	Jaccard
vegf_inhibitors	95.0	94.7	skip	FAIL	FAIL	FAIL
amd_biomarkers	91.7	91.7	skip	FAIL	FAIL	FAIL
dryamd_treatments	100	100	skip	FAIL	FAIL	FAIL
targeted_vs_general	88.0	88.0	skip	FAIL	FAIL	FAIL
imaging_vs_funct	50.0	66.7	skip	FAIL	FAIL	FAIL
genetic_vs_cellular	100	100	skip	FAIL	FAIL	FAIL
Runs OK / 45	45	45	0	0	0	0

returns Thing because the ontology lacks sufficient triples linking imaging methods to the biomarkers they measure.

The 270 sweep configurations in Table 6 (6 problems $\times$ 45 algorithm-metric-time-noise combinations) reveal four patterns. Algorithm choice dominates: CELOE succeeds on every configuration (108/108 OK across its two working metrics), while OCEL fails on every configuration (0/162 OK): it crashes at startup with a `NullPointerException` when loading our OWL/XML file, a known issue in DL-Learner 1.5 when the ontology declares `owl:disjointWith` axioms between named classes. Within CELOE, predicted accuracy and F-measure are equivalent: the two metrics produce identical top expressions on five of six problems and differ by 16 pp only on `imaging_vs_functional` (50% vs. 66.7%). Jaccard distance is structurally incompatible with CELOE because its `OEHeuristicRuntime` requires accuracy monotonicity under refinement, which Jaccard does not satisfy [23]. Finally, time and noise are non-discriminating: CELOE exhausts its search space within 30 s on 104 instances and 114 triples regardless of budget or noise setting. Scaling to the full PubMed AMD record ($\approx$ 25,000 abstracts) constitutes the natural next step.

5.3. Expert validation

The Age-related macular degeneration (AMD) Ontology is structured in seven categories: (1) Diseases, (2) Risk Factors, (3) Treatments, (4) Biomarkers, (5) Diagnostic Methods, (6) Clinical Outcomes and (7) Molecular Targets. The ontology correctly identifies DryAMD, WetAMD and GeographicAtrophy as AMD forms, all diagnosed by OCT imaging.

The ontology correctly identifies blindness as a symptom of WetAMD and GeographicAtrophy, though it should not apply to DryAMD where vision is only mildly impaired. Age is presented correctly by the ontology as possibly causing or indicating DryAMD, WetAMD and GeographicAtrophy and smoking as causing or increasing WetAMD and as the ontology evolves this could be added to the other forms as well.

Early Treatment Diabetic Retinopathy Study (ETDRS) is listed as a diagnostic method for DryAMD, though it requires further exploration: it may refer either to ETDRS charts for visual acuity or to the ETDRS grid used in OCT for macular thickness quantification, and its role in DryAMD diagnosis needs clarification.

Glaucoma and cataract are correctly listed as AMD comorbidities; uveitis requires further exploration as some studies consider it a risk factor [24] while others suggest patients with uveitis may be relatively spared [25].

Age is also listed as a risk factor for Retinopathy, but since this is a broad term, the ontology will need to be more specific as it evolves.

The Treatments category is separated into two subcategories: (1) Anti-Vascular Endothelial Growth Factor (anti-VEGF) therapy and (2) Other treatments. Ranibizumab, Aflibercept, Bevacizumab and Conbercept are correctly listed as anti-VEGF treatments with brand names Lucentis, Eylea and Avastin; Verteporfin is correctly listed for PDT treatment of ChoroidalNeovascularization. Other treatments include iSONEP (sonenpizumab), a monoclonal antibody targeting S1P [26] not yet FDA-approved, and Celecoxib, a COX-2 inhibitor still under research. Newer agents such as Faricimab (Vabysmo), a dual anti-VEGF-A and Angiopoietin-2 inhibitor, should be added in future iterations.

In the clinical outcomes category, signs and symptoms are together but could be further split into subcategories (6.1 Symptoms, 6.2 Signs); depression and dependence are better classified as outcomes linked through associatedWith. As the ontology evolves, symptoms could be subdivided into visual (reduced visual acuity, metamorphopsia [27]) and psychological (depression). Edema and scarring are correctly identified as clinical outcomes but would be more appropriately linked as signs of both ChoroidalNeovascularization and WetAMD.

While drusen is not present in the Clinical Outcomes category, it is correctly identified as a biomarker, the hallmark of AMD, and indication of DryAMD and GeographicAtrophy, though its presence could also indicate WetAMD and would benefit from being linked as a clinical outcome as well. As the ontology continues to evolve, drusen could be further divided into hard, soft or larger drusenoid RPE (Retinal Pigment Epithelium) detachment and additional elements could be added to the Clinical Outcomes category, such as pigment abnormalities ((RPE) hyperpigmentation and hypopigmentation), geographic atrophy, subretinal fluid, blood and fibrosis and fibrotic disciform scar [28].

Regarding relations, edema and hemorrhage are correctly linked to OCT imaging; OCT should also replace ICG for scarring visualization as the ontology evolves.

Overall, the majority of class assignments, disease forms, and treatment relations reviewed were judged clinically correct, with the flagged issues concentrated in a small number of recurring patterns: comorbidity versus risk-factor placement (e.g. uveitis, Retinopathy), symptom granularity for early-stage disease (e.g. blindness incorrectly applying to DryAMD), and category boundaries between biomarkers and clinical outcomes (e.g. drusen). This expert review was conducted as a qualitative clinical audit by a single ophthalmologist rather than a scored inter-rater evaluation; producing a quantitative agreement metric (e.g. per-assertion accept/revise/reject counts across multiple reviewers) is left as future work.

Data and Code Availability

Code, ontology, and configurations: <https://github.com/airi-utcn/AMD-ontology>; dataset: <https://doi.org/10.6084/m9.figshare.32301858>; demo: <https://calinaborzan-amd-ontology.hf.space/>; live ontology: <https://w3id.org/amd-ontology>.

6. Conclusion

This paper presented a fully agentic pipeline for constructing and maintaining an AMD OWL ontology from PubMed abstracts. Three LangChain agents handle literature discovery, knowledge extraction, and quality assurance, with a human reviewer approving every change. DL-Learner then mines the knowledge graph for non-trivial class definitions.

The community gains the following artefacts:

1. **AMD OWL ontology** (11 classes, 104 individuals, 114 triples; 38 HermiT-inferred axioms; 77–81% UMLS alignment, 20 pp above the pre-agentic baseline).
2. **Agentic pipeline** (literature discovery, extraction with delta validation, and ReAct-based ontology validation).
3. **DL-Learner experiment suite** (17 learning problems, 270-configuration parameter sweep; CELOE reaches 91.3% average accuracy and 100% on six problems): configurations and results included in the repository.
4. **Evaluation dataset** (UMLS alignment reports, competency-question SPARQL queries against 143 AMD patient questions [21], and pre-/post-agentic comparison): archived on Figshare [29].

Future work will scale the corpus to the full PubMed AMD record ($\approx 25,000$ abstracts), exercise DL-Learner’s heuristics under load, and integrate expert-annotated study types as non-trivial learning targets for class-expression learning.

Acknowledgments

This work was carried out at the Artificial Intelligence Research Institute AIRi@UTCN, Technical University of Cluj-Napoca.

Declaration on Generative AI

Calina Annemary Borzan has used Generative AI tools (Claude, Anthropic) in the preparation of this work for the following contributions, following the CEUR-WS GenAI Usage Taxonomy (<https://ceur-ws.org/GenAI/Taxonomy.html>):

- **Paraphrase and reword**: expressing ideas in different ways to ensure clarity and conciseness in the manuscript text
- **Improve writing style**: suggestions for sentence structure, word choice, and overall flow
- **Formatting assistance**: adapting the manuscript to the CEUR-ART formatting guidelines

In addition, outside the scope of the taxonomy above, the tool was used to assist with debugging and reviewing implementation code for the extraction pipeline described in this paper. No other author employed Generative AI tools in the preparation of this work. All AI-generated content was reviewed, verified, and edited by the authors, who take full responsibility for the accuracy and integrity of the final manuscript.

References

- [1] S. S. Sahoo, K. Kobow, J. Zhang, J. Buchhalter, M. Dayyani, D. P. Upadhyaya, K. Prantzas, M. Bhattacharjee, I. Blumcke, S. Wiebe, et al., Ontology-based feature engineering in machine learning workflows for heterogeneous epilepsy patient records, *Scientific reports* 12 (2022) 19430.
- [2] S. Sadruddin, J. D’Souza, E. Poupaki, A. Watkins, B. Karasulu, S. Auer, A. Mackus, E. Kessels, SCHEMA-MINERpro: Agentic AI for ontology grounding over LLM-discovered scientific schemas in a human-in-the-loop workflow, *Semantic Web Journal* (2025). To appear.

- [3] P. Mateiu, A. Groza, Ontology engineering with Large Language Models, in: 2023 25th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2023, pp. 226–229. doi:10.1109/SYNASC61333.2023.00038.
- [4] N. Fathallah, A. Das, S. De Giorgis, A. Poltronieri, P. Haase, L. Kovriguina, E. Simperl, A. Meroño-Peñuela, S. Staab, A. Algergawy, From GPT to Mistral: Cross-domain ontology learning with NeOn-GPT, *Semantic Web Journal* (2026). In press.
- [5] M. Val-Calvo, M. Egaña Aranguren, J. Mulero-Hernández, G. Almagro-Hernández, P. Deshmukh, J. A. Bernabé-Díaz, P. Espinoza-Arias, J. L. Sánchez-Fernández, J. Mueller, J. T. Fernández-Breis, OntoGenix: Leveraging Large Language Models for enhanced ontology engineering from datasets, *Information Processing & Management* 62 (2025) 104042. URL: <https://www.sciencedirect.com/science/article/pii/S0306457324004011>. doi:<https://doi.org/10.1016/j.ipm.2024.104042>.
- [6] H. Knublauch, D. Kontokostas, Shapes constraint language (SHACL), W3C Recommendation, 2017. <https://www.w3.org/TR/shacl/>.
- [7] J. Lehmann, DL-Learner: Learning Concepts in Description Logics, *Journal of Machine Learning Research* 10 (2009) 2639–2642.
- [8] C. Riefole, N. Fanizzi, C. d’Amato, Simple and interpretable probabilistic classifiers for knowledge graphs, 2024. URL: <https://arxiv.org/abs/2407.07045>. arXiv:2407.07045.
- [9] A. Lecu, A. Groza, L. Hawizy, Using LLMs and ontologies to extract causal relationships from medical abstracts, *Procedia Computer Science* 244 (2024) 443–452.
- [10] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, Z. Wang, HermiT: an OWL 2 reasoner, *Journal of automated reasoning* 53 (2014) 245–269.
- [11] A. Lecu, A. Groza, L. Hawizy, Reducing hallucinations in medical AI: A Knowledge Graph-Augmented Retrieval System for Evidence-Based Age-Related Macular Degeneration Information, *IEEE Access* 13 (2025) 210624–210639. doi:10.1109/ACCESS.2025.3643370.
- [12] A. Gilson, Bringing Large Language Models to Ophthalmology: Domain-Specific Ontologies and Evidence Attribution, Ph.D. thesis, Yale University, 2024.
- [13] M. R. Rezaei, et al., Agentic medical knowledge graphs enhance medical question answering: Bridging the gap between LLMs and evolving medical knowledge, *arXiv preprint arXiv:2502.13010* (2025). Available at: <https://arxiv.org/abs/2502.13010>.
- [14] C. J. Mungall, et al., The Monarch Disease Ontology (MONDO), *Nature Genetics* (2022).
- [15] L. M. Schriml, et al., Disease Ontology: a backbone for disease semantic integration, *Nucleic Acids Research* (2012).
- [16] B. Smith, et al., The OBO Foundry: coordinated evolution of ontologies, *Nature Biotechnology* (2007).
- [17] National Center for Biotechnology Information, Entrez Programming Utilities (E-utilities), <https://www.ncbi.nlm.nih.gov/books/NBK25501/>, 2023.
- [18] J. H. Caufield, et al., A change language for ontologies and knowledge graphs, *Database* (2025). doi:10.1093/database/baae133.
- [19] Y. Yang, G. Chen, B. He, Y. Zhao, Grounded knowledge graph extraction via LLMs: An anchor-constrained framework with provenance tracking, *Computers* 15 (2026). URL: <https://www.mdpi.com/2073-431X/15/3/178>. doi:10.3390/computers15030178.
- [20] U.S. National Library of Medicine, Medical Subject Headings (MeSH), <https://www.nlm.nih.gov/mesh/>, 2024.
- [21] G. A. Muntean, A. Marginean, A. Groza, I. Damian, S. A. Roman, M. C. Hapca, A. M. Sere, R. M. Mănoiu, M. V. Muntean, S. D. Nicoară, A qualitative evaluation of ChatGPT4 and PaLM2’s response to patient’s questions regarding age-related macular degeneration, *Diagnostics* 14 (2024) 1468.
- [22] M. A. Gargano, N. Matentzoglou, B. Coleman, et al., The Human Phenotype Ontology in 2024: phenotypes around the world, *Nucleic Acids Research* 52 (2024) D1333–D1346. doi:10.1093/nar/gkad1005.
- [23] J. Lehmann, P. Hitzler, Class expression learning for ontology engineering, *Journal of Web Semantics* 9 (2011) 71–81.

- [24] R.-H. Jing, D.-K. Zhou, Z.-Y. Yang, Z.-D. Ding, J.-H. Deng, Y. Xing, X.-F. Chen, Does uveitis increase the risk of age-related wet macular degeneration? a mendelian randomization study, *International Journal of Ophthalmology* 18 (2025) 1484.
- [25] A. Fox, C. A. Cukras, N. Bhatt, W. R. Tucker, R. B. Nussenblatt, H. N. Sen, Age-related macular degeneration among uveitis patients, *Investigative Ophthalmology & Visual Science* 55 (2014) 5323–5323.
- [26] M. Hanout, D. Ferraz, M. Ansari, N. Maqsood, S. Kherani, Y. J. Sepah, N. Rajagopalan, M. Ibrahim, D. V. Do, Q. D. Nguyen, Therapies for neovascular age-related macular degeneration: current approaches and pharmacologic agents in development, *BioMed research international* 2013 (2013) 830837.
- [27] D. Yorston, Age-related macular degeneration (amd): an introduction, *Community eye health* 37 (2025) 7.
- [28] S. Ruia, E. J. Kaufman, Macular degeneration, in: *StatPearls* [Internet], StatPearls Publishing, 2023.
- [29] C. A. Borzan, A. Lecu, G. Muntean, S. D. Nicoara, A. Groza, AMD Ontology – Data and Evaluation Results for Agentic Ontology Engineering using LLMs and DL-Learner, 2025. doi:10.6084/m9.figshare.32301858.