OntExtract: PROV-O Provenance Tracking for Document Analysis Workflows

Christopher B. Rauch
Drexel University
Philadelphia, PA, USA
cr625@drexel.edu

Hyung Wook Choi
Drexel University
Philadelphia, PA, USA
hc685@drexel.edu

Mat Kelly
Drexel University
Philadelphia, PA, USA
mrk335@drexel.edu

***Abstract*—Researchers have developed sophisticated methods for semantic change detection. These methods depend on foundational document processing operations like segmentation, entity extraction, and embedding generation, yet users face significant barriers when trying to combine these underlying tools into analytical workflows. We present OntExtract, a system that reduces these barriers through a unified interface for document processing workflows with integrated provenance tracking. PROV-O provenance concepts are embedded directly in the database schema. Each processing operation creates versioned outputs with corresponding provenance records. The system operates in two modes: API-enhanced mode uses large language models to orchestrate tool selection, while standalone mode relies on established NLP libraries (spaCy, NLTK, sentence-transformers). Users can apply different processing strategies to the same documents and compare results while maintaining clear provenance records. The PostgreSQL-based implementation with pgvector enables similarity searches and temporal filtering. This architecture enables reproducible semantic change analysis.**

***Index Terms*—document processing workflows, PROV-O provenance, reproducible analysis, NLP tool integration**

I. Introduction

Vocabulary in scholarly discourse is never static. Words shift in meaning across decades, across disciplines, and even within subfields of the same domain. This phenomenon of semantic change has long been recognized as both inevitable and problematic [1], [2]. Terminological shifts create semantic heterogeneity that acts as a major obstacle to interoperability between knowledge systems [3]. This becomes particularly acute when terms migrate across disciplines, where they can undergo substantial redefinition [4]. Without explicit mechanisms for reflection, scholars may fail to recognize when terms carry divergent assumptions across fields.

Multiple computational methods now exist for detecting semantic change. Methods using word embeddings track distributional shifts [5], structural analysis monitors ontological evolution [6], and contextual models capture fine-grained meaning variations [7]. However, the fragmentation of methods and multiplicity of available tools creates significant barriers to adoption. Users must invest substantial time learning different frameworks while maintaining incompatible development environments, each with distinct assumptions and limitations [2], [8].

Beyond the challenge of mastering individual tools is the even deeper challenge of synthesizing diverse analytical outputs into coherent interpretations. Each tool produces results in its own format, incorporating its own assumptions and limitations [2], [8]. Researchers must not only run multiple analyses but also reconcile distributional measurements with structural observations, integrate statistical patterns with categorical boundaries, and ultimately construct unified interpretations from methodologically distinct approaches. This synthesis burden multiplies when tracking semantic evolution across time periods or disciplines, where different analytical frameworks may be optimal for different contexts.

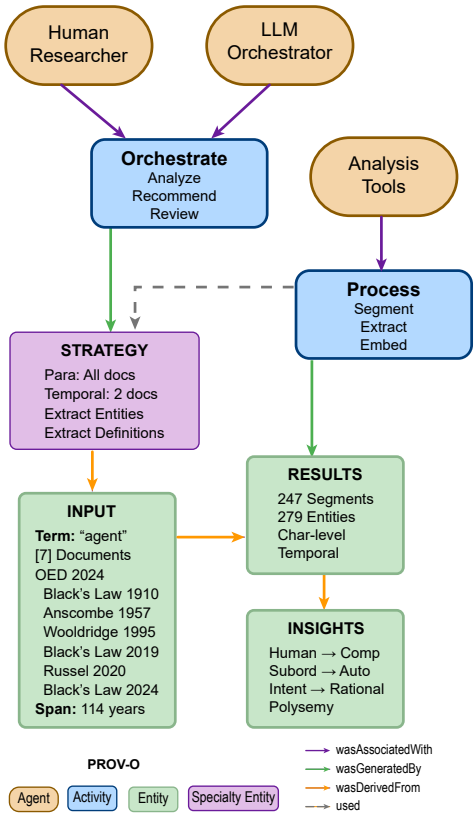

Fig. 1. PROV-O provenance architecture demonstrated through semantic evolution analysis of "agent" across seven documents (1910-2024) spanning law, philosophy, and AI. The LLM orchestrator analyzes documents, recommends processing strategies, and coordinates analysis tools. PROV-O relationships (wasAssociatedWith, wasGeneratedBy, wasDerivedFrom, used) maintain queryable provenance throughout the workflow.

OntExtract explores the use of large language models as integration engines for multiple analytical methods. The system builds on recent work in tool-augmented language models [9] and coordinates different analytical pipelines based on document characteristics and research requirements to reduce coordination challenges [2], [8].

PROV-O, the W3C standard for representing provenance information, provides a formal vocabulary for describing the origins and history of computational artifacts [10]. The standard models provenance through agents (who performed actions), activities (what was done), and entities (what was created or modified). This formal structure enables reproducible research by documenting the complete lineage of analytical results. The implementation incorporates PROV-O provenance concepts directly into the database structure, adapting the standard vocabulary for LLM orchestration similar to how D-PROV [11] extends PROV for scientific workflows. This design captures analytical provenance, including which tools were selected and how orchestration decisions incorporated researcher feedback.

A. Related Work

Missier et al. [11] demonstrate how the W3C PROV-O standard enables computational reproducibility in scientific workflows. Their D-PROV extension provides a model to adapt PROV-O to specific computational domains. Jatowt and Duh [12] proposed an early framework for analyzing semantic change across time, introducing multiple perspectives on semantic shift in diachronic corpora. Dubossarsky et al. [13] introduced Temporal Referencing, which reduces noise in semantic shift models by relabeling target words with time-specific information.

Kutuzov et al. [1] surveyed diachronic word embeddings, emphasizing the need for interpretability and reproducibility. They noted that differences in corpus preprocessing, embedding training, and alignment procedures can significantly affect outcomes. Montariol et al. [14] proposed scalable neural methods and emphasized interpretable methods where the derivation process is as transparent as the results. Recent developments in LLM orchestration [9] position language models as agents capable of complex problem-solving and decision-making. However, provenance tracking for analytical decisions in LLM-orchestrated workflows remains underdeveloped. OntExtract makes provenance a first-class concern through direct integration of PROV-O concepts into the database architecture.

II. Method

A. System Architecture and Document Processing

OntExtract implements a modular architecture that combines structured document extraction with provenance-aware analysis pipelines. The system operates in two modes based on available resources. Standalone mode provides core document processing through established NLP libraries (spaCy for named entity recognition, NLTK for tokenization, sentence-transformers for semantic embeddings) without requiring external API access. API-enhanced mode augments this foundation with LLM-orchestrated tool selection using Claude (Anthropic) or GPT (OpenAI) to analyze document characteristics and recommend appropriate processing strategies. The architecture supports multiple LLM backends through an abstraction layer. All processing operations use the same underlying NLP tools regardless of mode; the distinction lies in how tools are selected and coordinated.

1) Document Processing Strategies: The system integrates multiple NLP libraries for document analysis. Users select from three primary segmentation approaches through the interface. Paragraph-based segmentation uses regular expression patterns to identify natural text boundaries. Sentence-level segmentation uses NLTK tokenizers for fine-grained analysis. Semantic segmentation applies sentence-transformers to identify meaning-based boundaries and groups sentences by topic similarity.

Entity extraction uses pretrained models from spaCy to identify named entities, locations, organizations, and temporal markers. The system preserves character-level position information for extracted elements. Embedding generation uses sentence-transformers to create vector representations of text segments for semantic similarity searches through the pgvector-enabled PostgreSQL database

B. LLM Orchestration Mechanism

The API-enhanced mode implements a five-stage orchestration workflow that coordinates tool selection and execution. The workflow uses LangGraph, a framework for building stateful multi-agent applications with LLMs. LangGraph manages state transitions and decision points through a graph-based architecture where each stage maintains explicit state and can conditionally branch based on analysis results.

a) Experiment Analysis: The configured LLM backend (currently Claude, with support for other providers through the abstraction layer) analyzes experiment goals and document characteristics to identify focus terms and research objectives. The analysis examines document metadata and user-specified research questions to produce structured experiment context.

b) Strategy Recommendation: Using structured context from the previous stage, the LLM recommends specific tools for each document and generates a processing strategy with reasoning for each selection and confidence scores. For example, it might recommend entity extraction for historical documents with many proper nouns, or semantic segmentation for modern technical papers with complex topic boundaries.

c) Human Review: Users examine LLM recommendations through an interface that displays tool selections with reasoning and confidence scores. They can approve recommendations, modify tool selections, or add processing notes. The system records all review decisions as part of the provenance trail, and the workflow conditionally branches based on whether modifications are requested.

d) Strategy Execution: In the final stage, results across all processed documents are analyzed to generate cross-document insights and comparative summaries. For experiments with focus terms, term evolution analysis is performed across the

document set. All synthesis decisions become part of the queryable provenance record.

e) Standalone Mode Operation: Without API access, users manually select tools through the interface. Manual selections are recorded with the same provenance structure to maintain consistency across modes. Core processing capabilities use the same NLP tools regardless of mode.

C. Document Versioning and Processing Integrity

OntExtract creates a new document version for every processing operation while preserving originals unchanged. This design maintains document integrity and enables iterative analysis with comparative evaluation of different processing strategies.

The versioning system maintains three document types: originals (permanently unmodified), processed documents (from standalone operations), and experimental documents (from structured experiments). Each type maintains metadata including version identifiers, processing notes, and source document references. When users apply a processing operation, a new document is created and linked to the original through the source document identifier. Text segments, embeddings, and analytical outputs attach to these processed documents. These versioned documents become entities in the PROV-O provenance model.

D. PROV-O Database Architecture

The PROV-O implementation captures complete provenance for these versioned documents through two complementary database architectures. Document processing operations (upload, extraction, segmentation, embedding generation) use W3C PROV-O compliant tables with strict relational constraints. Experiment orchestration state (LLM recommendations, human review decisions, execution traces) applies PROV-O concepts to workflow data stored in JSONB fields for flexible schema evolution.

The PROV-O tables record three core elements. Agents include human researchers and analysis tools with version metadata. Activities include document upload, text extraction, segmentation, and embedding generation. Entities include document versions with character-level position tracking. Four relationships enable workflow reconstruction: `wasDerivedFrom` links document versions to their sources, `wasGeneratedBy` connects outputs to generating processes, `used` records which entities were consumed by activities, and `wasAssociatedWith` maps operations to specific tool versions (such as spaCy 3.8.3).

Researchers can trace any analytical output to its generating process and reconstruct complete processing histories through derivation chains, identifying which tool version produced specific results. This approach aligns with recent calls for interpretable semantic change detection methods [14], where the derivation process is as important as the results themselves.

E. Reproducibility and Settings Management

Centralized settings management ensures reproducibility through eighteen system-wide configuration parameters that can be set before processing and are automatically recorded when experiments run. Parameters include model selections (e.g., `spacy_model`, `embedding_model`), processing methods, and output dimensions. The database stores these settings with version history, and experiments capture their complete configuration state at creation time.

The system distinguishes between deterministic and non-deterministic operations. Document processing operations (segmentation, extraction, embedding generation) produce identical outputs given identical settings and tool versions. LLM orchestration recommendations vary across runs due to model non-determinism, but the system records complete decision context including reasoning, confidence scores, and human modifications. Version pinning of all tools and parameters enables exact reproduction of the document processing pipeline, while captured decision provenance supports understanding and evaluation of orchestration strategies.

F. Implementation and Validation

The current implementation of OntExtract is available as open source at https://github.com/MatLab-Research/OntExtract and builds on OntServe (https://github.com/MatLab-Research/OntServe) for ontology management capabilities. The prototype runs on a single-CPU server and makes direct API calls to the Anthropic API for LLM orchestration. This infrastructure configuration limits concurrent processing and throughput.

The prototype demonstrates core architectural concepts including document versioning, PROV-O provenance tracking, dual-mode processing, five-stage LLM orchestration, and human-in-the-loop feedback mechanisms. Performance characteristics reflect this minimal infrastructure deployment. Significantly higher throughput could be achieved with access to institutional compute clusters, parallel processing across multiple workers, distributed task scheduling, and optimized API request batching. The current implementation focuses on single-term analysis to establish the methodology. The modular architecture supports expansion to more complex semantic phenomena as development continues.

Entity extraction in standalone mode uses spaCy's pretrained models (en_core_web_sm) with pass-through results from the underlying library. Performance characteristics match spaCy's published benchmarks on standard datasets including CoNLL-2003. Domain-specific texts may require fine-tuned models for production use.

The document versioning and provenance tracking system adds minimal storage overhead (typically under 5%). JSONB compression in PostgreSQL reduces provenance record sizes. The pgvector extension indexes high-dimensional embeddings (384 dimensions for MiniLM-L6-v2) with minimal query latency impact.

G. Case Study: Agent Evolution

Figure 2 demonstrates the complete PROV-O architecture through semantic analysis of "agent" across seven documents

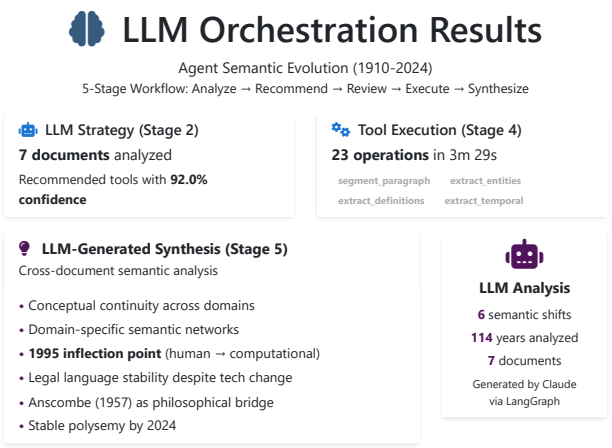

Fig. 2. LLM orchestration results: (Stage 2) 92% confidence strategy across 7 documents, (Stage 4) 23 tool executions, (Stage 5) synthesis summary of 6 semantic shifts in "agent" (1910-2024).

spanning 1910-2024: legal dictionaries (Black's Law Dictionary 1910, 2019, 2024), philosophical work (Anscombe 1957), AI foundations (Wooldridge & Jennings 1995, Russell & Norvig 2020), and lexicography (OED 2024).

The LLM orchestrator analyzes the 114-year temporal span and cross-disciplinary scope, identifying the research objective as tracking conceptual migration of "agent" from legal representation through philosophical agency to computational autonomy. Paragraph segmentation, entity extraction, and definition identification are recommended for all documents, with temporal extraction added for the two most comprehensive sources. Strategy confidence reaches 0.92, reflecting high structural clarity across the corpus.

After researcher approval, processing coordinates spaCy, NLTK, and sentence-transformers through 23 tool executions. The synthesis identifies six semantic evolution patterns: conceptual continuity of purposeful action across domains, domain-specific elaboration within specialized semantic networks, temporal stratification with 1995 marking the computational inflection point, technological pressure on legal language while maintaining definitional stability, philosophical mediation through Anscombe's intentional action framework, and polysemous stabilization enabling distinct but related meanings in law, philosophy, and AI.

The complete PROV-O chain enables reproducibility through queryable records of which tools processed each document, when operations occurred, and why specific tools were selected for each source.

III. Discussion and Conclusion

OntExtract addresses the fragmentation of semantic change detection tools through a unified interface for document processing workflows. Dual-mode operation supports manual tool selection through established NLP libraries or LLM-orchestrated processing with human review. PROV-O provenance tracking maintains complete records of analytical decisions.

The provenance implementation records each processing operation as database entities with timestamps and configuration metadata including tool versions and parameters. This enables tracing which methods processed each document and reconstructing complete analytical workflows. Incremental adoption is supported through standalone mode deployment with optional LLM orchestration when API access becomes available.

The current prototype demonstrates core architectural concepts through single-term analysis. Entity extraction accuracy in standalone mode relies on pretrained models from spaCy with performance matching published benchmarks for general domain texts. Domain-specific applications may require fine-tuned models, and users should validate extraction results for their specific use cases.

Limitations include dependence on commercial LLM APIs for full orchestration, manual processing in standalone mode, English-language focus, and restriction to individual term analysis. Future work will integrate locally-hosted open-source LLMs to reduce API dependence and expand to phrase-level semantic analysis, with validation of orchestration reliability through comparison with expert annotations and user studies.

References

[1] A. Kutuzov *et al.*, "Diachronic word embeddings and semantic shifts: a survey," in *Proc. COLING*, 2018.
[2] N. Tahmasebi *et al.*, "Survey of computational approaches to lexical semantic change detection," in *Computational Approaches to Semantic Change*. Language Science Press, 2021.
[3] M. Maree and M. Belkhatir, "Addressing semantic heterogeneity through multiple knowledge base assisted merging of domain-specific ontologies," *Knowledge-Based Systems*, vol. 73, pp. 199–211, 2015.
[4] S. H. Frost and P. M. Jean, "Bridging the disciplines: Interdisciplinary discourse and faculty scholarship," *The Journal of Higher Education*, vol. 74, no. 2, pp. 119–149, 2003.
[5] W. L. Hamilton *et al.*, "Diachronic word embeddings reveal statistical laws of semantic change," in *Proc. ACL*, 2016.
[6] T. G. Stavropoulos *et al.*, "SemaDrift: A hybrid method and visual tools to measure semantic drift in ontologies," *Journal of Web Semantics*, vol. 54, pp. 87–106, 2019.
[7] M. Giulianelli *et al.*, "Analysing lexical semantic change with contextualised word representations," in *Proc. ACL*, 2020.
[8] S. Hengchen *et al.*, "Challenges for computational lexical semantic change," in *Computational Approaches to Semantic Change*. Language Science Press, 2021.
[9] T. Guo *et al.*, "Large language model based multi-agents: A survey of progress and challenges," in *Proc. IJCAI*, 2024.
[10] W3C, "PROV-O: The PROV ontology," 2013, w3C Recommendation.
[11] P. Missier *et al.*, "D-PROV: Extending the PROV provenance model with workflow structure," in *Proc. TaPP*, 2013.
[12] A. Jatowt and K. Duh, "A framework for analyzing semantic change of words across time," in *Proc. JCDL*, 2014.
[13] H. Dubossarsky *et al.*, "Time-out: Temporal referencing for robust modeling of lexical semantic change," in *Proc. ACL*, 2019.
[14] S. Montariol *et al.*, "Scalable and interpretable semantic change detection," in *Proc. NAACL*, 2021.