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
mkelly@drexel.edu

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

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

I. INTRODUCTION

Vocabulary in scholarly discourse is never static. Words shift in meaning across time and disciplines, with changes occurring 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 problem 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. Distributional approaches using word embeddings track meaning shifts over time [5], while structural analysis monitors ontological evolution [6] and contextual models capture fine-grained variations [7]. Yet, the fragmentation of these methods and the multiplicity of available tools create adoption challenges. Users must invest substantial time learning different frameworks while maintaining incompatible development environments, each with distinct requirements and limitations [2], [8]. The synthesis burden multiplies when reconciling distributional measurements with structural obser-
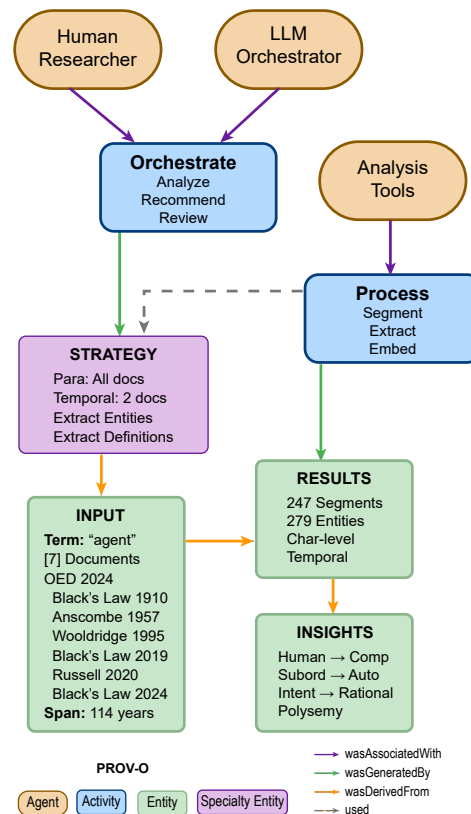

Fig. 1. PROV-O provenance architecture demonstrated through semantic evolution analysis of "agent" across seven documents (1910-2024) spanning law, philosophy, and AI. The LLM orchestrator analyzes documents, recommends processing strategies, and coordinates analysis tools. PROV-O relationships (wasAssociatedWith,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

vations and constructing unified interpretations from methodologically distinct approaches.

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 as entities (artifacts created or modified) and the activities and agents responsible for their generation. 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. Dubossarsky et al. [13] introduced Temporal Referencing to reduce noise in semantic shift models.

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, NLTK, sentence-transformers) without requiring external API access. API-enhanced mode augments this foundation with LLM-orchestrated tool selection using Claude (Anthropic) 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.

The system maintains a term management database with definitions from authoritative sources (Oxford English Dictionary, Merriam-Webster, and other lexical databases) and user-defined entries. Each term includes context anchors for disambiguation and supports versioning to track definition changes over time. Term definitions and context anchors are incorporated into LLM prompts during orchestration to provide semantic context.

1) Document Processing Strategies: The system integrates multiple NLP libraries for document analysis. Users select segmentation approaches through the interface. Paragraph-based segmentation uses regular expression patterns to identify natural text boundaries, while 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 spaCy models 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. Available tools include `extract_entities_spacy`, `extract_temporal`, `extract_definitions`, and `semantic_similarity`.

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. The LLM receives structured prompts containing document metadata and term definitions with context anchors from the database. The system uses few-shot prompting with exemplar tool selections for common document types. LLM responses are parsed into structured JSON containing tool recommendations, confidence scores (0-1 scale), rationale for choice, and execution parameters for each tool selection.

a) Experiment Analysis: The configured LLM backend 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: The LLM uses structured context from the previous stage to recommend specific tools for each document and generate a processing strategy with reasoning for each selection and confidence scores. For historical documents with many proper nouns, entity extraction might be recommended, while semantic segmentation might be suggested for modern technical papers with complex topic boundaries.

c) Human Review: Users examine LLM recommendations through an interface that displays tool selections, along 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 analytical trail, and the workflow conditionally branches based on whether modifications are requested.

d) Strategy Execution: Approved strategies are executed using local NLP tools. The system processes documents in parallel, applying selected tools without LLM involvement at this stage. Each tool execution creates ProcessingArtifact records with structured outputs and character-level position tracking.

e) Results Synthesis: The LLM analyzes processing outputs from all documents to identify patterns and organize findings. For temporal evolution experiments, the system generates structured term cards that display frequency data and co-occurrence patterns for each time period. The synthesis organizes but does not interpret results, preserving researcher authority over analytical conclusions.

f) Standalone Mode Operation: In standalone mode, 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 across modes.

C. Processing Artifacts and Document Integrity

OntExtract preserves original documents unchanged. Analysis results are stored as ProcessingArtifacts (separate database entities linked to source documents through PROV-O relationships). This design maintains document integrity and enables the application of multiple processing strategies to identical sources for comparative evaluation.

The ProcessingArtifact table implements these provenance entities and stores the operation type, timestamps, configuration parameters, and results. Artifacts include text segments with character positions, entity extractions with confidence scores, embedding vectors, and structured analytical outputs. Each artifact references the source document through PROV-O relationships.

D. PROV-O Database Architecture

The PROV-O implementation captures complete provenance for processing artifacts. Document processing operations use W3C PROV-O compliant tables with strict relational constraints. The experiment orchestration state uses PROV-O concepts for workflow data stored in JSONB fields, enabling flexible schema evolution alongside LLM recommendations and human review decisions.

The implementation applies the PROV-O entity-activity-agent model directly to the database structure. Human researchers and analysis tools are recorded as agents with version metadata. Activities include document upload and various text processing operations. Entities include artifacts with character-level position tracking. Four relationships enable workflow reconstruction: `wasDerivedFrom` links artifacts to source documents, `wasGeneratedBy` connects artifacts to generating processes, `used` records which entities were consumed by activities, and `wasAssociatedWith` maps operations to specific tool versions (such as spaCy 3.8.11).

This architecture enables researchers to trace any artifact to the generating process and reconstruct complete processing histories through queryable provenance chains that identify which tool version produced specific results. The derivation process becomes as transparent as the results themselves, aligning with recent calls for interpretable semantic change detection methods [14].

E. Reproducibility and Settings Management

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

The system distinguishes between deterministic and non-deterministic operations. Document processing operations (segmentation and extraction) produce identical outputs given identical settings and tool versions. LLM orchestration recommendations vary across runs due to model non-determinism, but the system records the complete decision context for each recommendation. Version pinning of all tools and parameters enables exact reproduction of the document processing pipeline, while captured decision provenance supports the 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 focuses on single-term analysis to establish the methodology.

Entity extraction uses pretrained models from spaCy (en_core_web_sm) with pass-through results from the underlying library. Extraction accuracy depends on domain alignment with training corpora. Technical and historical texts often require domain-specific fine-tuning for optimal performance. Users should validate extraction results for their specific use cases and consider the confidence scores provided with each extracted entity.

The ProcessingArtifact and provenance tracking system add minimal storage overhead (typically under 5%). JSONB compression in PostgreSQL reduces the size of provenance records. The pgvector extension indexes high-dimensional embeddings with minimal impact on query latency.

Processing performance demonstrates the feasibility of the architecture, but it is not optimized for production deployment. Document processing operations complete in reasonable time-frames for research workflows, with most orchestration time spent in API communication rather than local computation.

G. Case Study: Agent Evolution

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

LLM Orchestration Results

Agent Semantic Evolution (1910-2024)
5-Stage Workflow: Analyze → Recommend → Review → Execute → Synthesize

LLM Strategy (Stage 2)
7 documents analyzed
Recommended tools with **92.0% confidence**

Tool Execution (Stage 4)
23 operations in 3m 29s
segment_paragraph   extract_entities
extract_definitions   extract_temporal

LLM-Generated Synthesis (Stage 5)
Cross-document semantic analysis

- Conceptual continuity across domains
- Domain-specific semantic networks
- **1995 inflection point** (human → computational)
- Legal language stability despite tech change
- Anscombe (1957) as philosophical bridge
- Stable polysemy by 2024

LLM Analysis
6 semantic shifts
114 years analyzed
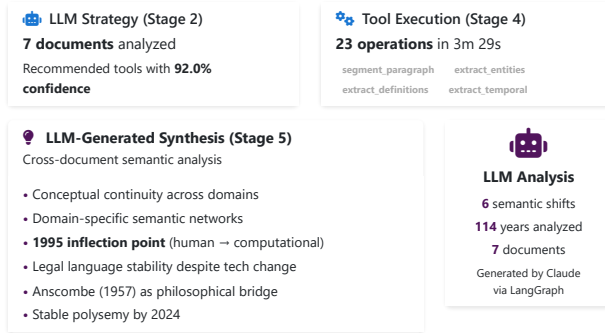**7** documents
Generated by Claude via LangGraph

Fig. 2. LLM orchestration interface showing results from the "agent" case study. The interface displays strategy recommendation (Stage 2) with 92% confidence, tool execution details (Stage 4), and structured synthesis output (Stage 5) organizing six semantic evolution patterns across the 114-year document corpus.

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 the conceptual migration of "agent" from legal representation through philosophical agency to computational autonomy. Paragraph segmentation with extract_entities_spacy and extract_definitions is recommended for all documents, with extract_temporal added for the two most comprehensive sources. Strategy confidence reaches 0.92, reflecting high structural clarity across the corpus.

After researcher approval, the tool operations are executed using the established NLP libraries. The synthesis identifies six semantic evolution patterns: conceptual continuity across domains, domain-specific elaboration within specialized networks, temporal stratification with 1995 marking the computational inflection, definitional stability in legal language despite technological pressure, philosophical mediation through Anscombe's framework, and polysemous stabilization enabling distinct meanings across fields. The complete PROV-O chain enables reproducibility through queryable records of tool selection and execution provenance.

III. Discussion and Conclusion

OntExtract addresses the fragmentation of semantic change detection tools by providing a unified interface for document analysis workflows. Dual-mode operation supports manual tool selection through established NLP libraries or LLM-orchestrated processing with human review. PROV-O provenance tracking enables researchers to trace analytical decisions and reproduce processing workflows with complete

provenance for tools and parameters. 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, with entity extraction accuracy dependent on domain-specific model selection.

Limitations include reliance on commercial LLM APIs for full orchestration, manual processing in standalone mode, an English-language focus, and a 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 comparisons with expert annotations and user studies.

REFERENCES

[1] A. Kutuzov, L. Øvrelid, T. Szymanski, and E. Velldal, "Diachronic word embeddings and semantic shifts: a survey," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 1384–1397. [Online]. Available: https://aclanthology.org/C18-1117/

[2] N. Tahmasebi, L. Borin, and A. Jatowt, "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, J. Leskovec, and D. Jurafsky, "Diachronic word embeddings reveal statistical laws of semantic change," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1489–1501.

[6] T. G. Stavropoulos, S. Andreadis, E. Kontopoulos, and I. Kompatsiaris, "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, M. Del Tredici, and R. Fernández, "Analysing lexical semantic change with contextualised word representations," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 3960–3973.

[8] S. Hengchen *et al.*, "Challenges for computational lexical semantic change," in *Computational Approaches to Semantic Change*. Language Science Press, 2021.

[9] T. Guo, X. Chen, Y. Wang, R. Chang, S. Pei, N. V. Chawla, O. Wiest, and X. Zhang, "Large language model based multi-agents: A survey of progress and challenges," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2024, pp. 8048–8057.

[10] W3C, "PROV-O: The PROV ontology," 2013, w3C Recommendation.

[11] P. Missier, S. Dey, K. Belhajjame, V. Cuevas-Vicenttín, and B. Ludäscher, "D-PROV: Extending the PROV provenance model with workflow structure," in *5th USENIX Workshop on the Theory and Practice of Provenance (TaPP 13)*, 2013. [Online]. Available: https://www.usenix.org/conference/tapp13/technical-sessions/presentation/missier

[12] A. Jatowt and K. Duh, "A framework for analyzing semantic change of words across time," in *IEEE/ACM Joint Conference on Digital Libraries*, 2014, pp. 229–238.

[13] H. Dubossarsky, S. Hengchen, N. Tahmasebi, and D. Schlechtweg, "Time-out: Temporal referencing for robust modeling of lexical semantic change," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 457–470.

[14] S. Montariol, M. Martinc, and L. Pivovarova, "Scalable and interpretable semantic change detection," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 4642–4652.