AgriQuant: An ontology-backed event corpus for reproducible event studies in animal-health and agribusiness equities

by

Jarrar Amjad

B.E., National University of Sciences and Technology, 2018

A THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computer Science
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2026

Approved by:

Major Professor
Hande Küçük McGinty

Copyright

© Jarrar Amjad 2026.

Abstract

An event study asks how a company's stock moved around a piece of news, such as a drug approval or an acquisition. The statistics are well established, but the list of events behind them is often hard to check, and a wrong date or a duplicated record can change the result even when the code is correct. This research presents AgriQuant, a knowledge-graph-backed system that keeps financial events for animal-health and agribusiness companies in a form that can be checked, traced, and rerun. The events are encoded as a Resource Description Framework knowledge graph, the Web Ontology Language defines the kinds of events, and the Shapes Constraint Language checks that every record is complete. Each event keeps two dates, when it happened and when the public first learned about it, and only events with a confirmed public date are analyzed. A review of primary sources* gave every collected record a final decision and produced a corpus of 162 events across 18 companies. A case study applies a standard market-model event study to the eight animal-health stocks among them from 2021 to 2025, using a fixed snapshot of daily closing prices from Stooq. Rerunning the analysis reproduces the saved results exactly, every result lists the events behind it, and the sensitivity checks show how much the sample and the results depend on which events count. The outcome of this research is not a stock forecast but a reproducible link from each reviewed event and its date evidence to the exact data and result it produced, which makes event-study results traceable and explainable.

*The review was carried out with artificial intelligence (AI) assistance, as described in the Preface.

Table of Contents

List of Figures viii

List of Tables ix

Acknowledgments x

Preface xi

1 Introduction 1

1.1 Research objective 3

1.2 Research questions 3

1.3 Contributions 4

1.4 Summary of findings 5

1.5 Organization 5

2 Background and related work 7

2.1 Event-study methodology 7

2.2 Representing events as data 8

2.3 Forecasting and machine learning 10

2.4 Ontology engineering and reproducible research 10

3 Methodology and ontology design 12

3.1 Research methodology 12

3.2 Design requirements 14

3.3 Core classes 15

3.4 Required and optional properties . . . 16
3.5 Identity and authoritative fields . . . 17
3.6 OWL semantics and SHACL validation . . . 19
3.7 Directional annotation . . . 20

4 Corpus construction and source review . . . 21
4.1 Corpus scope . . . 21
4.2 Why a source link is not enough . . . 22
4.3 Review protocol and primary sources . . . 22
4.4 Review outcome . . . 23
4.5 Event dates and disclosure dates . . . 26
4.6 Building the Turtle file . . . 26
4.7 What the review changed for the analysis . . . 27

5 System and reproducibility architecture . . . 28
5.1 Architecture overview . . . 28
5.2 Notebooks . . . 28
5.3 Reading events from the graph . . . 29
5.4 Shared event selection . . . 30
5.5 Market snapshot . . . 30
5.6 Analysis identity . . . 30
5.7 Generated outputs . . . 31
5.8 Web application . . . 31

6 Event-study design and results . . . 33
6.1 Evaluation protocol . . . 33
6.2 Return and market model . . . 34
6.3 Analysis units and dependence . . . 36
6.4 Circular-shift placebo check . . . 36

6.5 RQ1: rule conformance . . . 37
6.6 Competency questions . . . 38
6.7 RQ2: date evidence and eligibility . . . 39
6.8 Case-study results . . . 40
6.9 What the review changed in the results . . . 42
6.10 RQ3: repeatable output . . . 43
6.11 RQ4: sensitivity to event definition . . . 43
6.12 Result summary . . . 44

7 Discussion and conclusions . . . 45
7.1 Answers to the research questions . . . 45
7.2 What "ontology backed" means here . . . 46
7.3 How reviewed inputs shape the result . . . 47
7.4 Threats to validity . . . 47
7.4.1 Corpus coverage and selection . . . 47
7.4.2 Event timing . . . 48
7.4.3 Other news and causation . . . 48
7.4.4 Market data . . . 48
7.4.5 Statistical power and multiple comparisons . . . 49
7.4.6 Review and annotation . . . 49
7.4.7 Ontology evaluation and persistence . . . 49
7.5 Lessons for research and engineering . . . 50
7.6 Future work . . . 50
7.7 Conclusion . . . 52

Bibliography . . . 53

A Reproduction procedure . . . 61
A.1 Install dependencies . . . 61

A.2 Run the notebooks 61
A.3 Verify the market snapshot 62
A.4 Regenerate analysis output 63
A.5 Check the web application 63
A.6 Build the documents 64
A.7 Reproducibility boundary 64

B Event data contract 65
B.1 Controlled event types 66
B.2 Eligibility state model 68
B.3 Analysis-count definitions 68

C Methodological safeguards and rejected practices 69
C.1 Interpretive consequence 72

D Competency-question queries 73

List of Figures

3.1 End-to-end AgriQuant workflow 14

3.2 OntoGraf view of the AgriQuant event schema in Protégé Desktop 18

4.1 The May 5, 2023 FDA approval of Librela and its connections in Protégé OntoGraf 25

List of Tables

2.1 Purposes and data controls of representative event-data approaches 9

3.1 KNARM steps as applied in this research 13

3.2 The six AgriQuant event classes 15

3.3 Core AgriQuant event properties 16

4.1 From original records to analyzed events 24

4.2 Final events and disclosure-date eligibility 24

5.1 The three AgriQuant notebooks 29

6.1 The eight stocks in the animal-health case study 34

6.2 What each competency-question query returns 38

6.3 Descriptive own-firm CAR in percentage points 40

6.4 Circular-shift diagnostics for mean CAR$[0, +10]$ in percentage points 40

6.5 Placebo p_{shift} for every company and CAR window 41

6.6 Case-study results with and without the review 42

6.7 Sensitivity to ontology-backed subset definitions 44

B.1 Analysis-facing event fields 65

B.2 Controlled event types and their expected classes 67

B.3 Analysis-count definitions 68

C.1 Safeguard ledger 69

Acknowledgments

I thank Dr. Hande Küçük McGinty for advising this research and for guidance on ontology engineering and research design. I thank Dr. Allen M. Featherstone and Dr. Doina Caragea for serving on my committee, and Dr. Featherstone also for guidance on agricultural-finance framing and for identifying relevant prior work. I also thank members of the KONCORDANT Lab at Kansas State University for discussions that helped refine the project.

Preface

Use of generative artificial intelligence

Large language model (LLM) assistance was used in the data-collection stage of this research. It supported discovery of the initial candidate list of events. The researcher reviewed the source, date, classification, and inclusion decisions for every original record with artificial intelligence (AI) assistance from Codex, OpenAI's coding assistant, and validated the final records and shapes (Chapters 3 and 4). Codex and Claude Code, Anthropic's coding assistant, also assisted software development and manuscript editing. The author remains responsible for the research claims and their verification.

Chapter 1

Introduction

When a company announces news, such as a regulatory approval or an acquisition, its stock price may react. An event study measures that reaction. It estimates what the stock would normally have returned and then looks at how far the actual return differed around the event date. The method is standard in economics and finance, and its equations are well documented.[1–3]

The harder part is the list of events. Someone has to decide what happened, which company it belongs to, when the public learned about it, and which source supports each of those decisions. That list is rarely kept in a form others can check, and small mistakes in it pass straight through correct statistical code.

Animal health was chosen as the use case for this research because its news is varied: veterinary drug approvals, label expansions, clinical and development studies, safety warnings, product launches, acquisitions, plant expansions, and production problems. These different kinds of news all feed the same decision in an event study: which public events should be matched to a company's stock returns, and on which date. They also carry different information. An approval has a regulator and usually one or more target species, while an acquisition has neither. Knowledge graphs are designed for this kind of problem. A knowledge graph represents entities and the named relationships between them, so that related

facts spread across different sources can be connected and queried together.[4, 5]

In addition to drawing on many kinds of sources, an event list is prone to ordinary transcription and classification errors. A record dated only to the month gets the first day of that month as if it were exact. Two records describe the same announcement and are counted twice. A product web page shows the date it was last edited instead of the date of the original announcement. An approval is filed as a different kind of event, category names drift as a spreadsheet is edited, and a reported result loses track of the rows that produced it. None of these is a statistics problem, but each one changes the statistics.

This research presents *AgriQuant*, a knowledge-graph-backed system for defining and checking financial events for animal-health and agribusiness companies. Agribusiness covers the firms that supply farms with equipment, fertilizer, seeds, and crop chemicals and the firms that store, trade, and process farm commodities.[6] The events are stored in the Resource Description Framework (RDF), the World Wide Web Consortium's standard model for graph data, and written in Turtle, a human-readable text format for RDF kept in `.ttl` files.[7, 8] An ontology is an explicit specification of a conceptualization: it names the concepts of a domain and states how they relate to each other.[9] The Web Ontology Language (OWL), a standard language for writing ontologies, defines the kinds of events in AgriQuant and how they relate.[10] The Shapes Constraint Language (SHACL) works as a data checker: it describes the shape every stored record must have, such as its required fields and allowed values, and reports any record that does not fit.[11] This research followed the Knowledge Acquisition and Representation Methodology (KNARM), a nine-step method for building an ontology around a use case together with domain experts, as described in Chapter 3.[12, 13]

During this research it became clear that the date an event happened often differs from the date it was first made public. A regulator may approve a product on one day and announce it days later, and a web page may show only a later update date. To keep the two apart, AgriQuant records every collected event in a *verification ledger*: a structured file, written in JavaScript Object Notation (JSON),[14] that stores for each record the evidence,

the review decision, the reason for it, and both dates. Only events with a confirmed public date enter the event study. Chapter 4 describes the ledger in detail.

The case study uses a fixed copy of daily closing prices from the Stooq historical market-data service.[15] The file's SHA-256 hash, a digital fingerprint of the file, is checked before every analysis, so the prices cannot change between runs.[16] The analysis is written as three Jupyter notebooks, documents that combine explanatory text, program code in the Python language, and the output of that code, so that each step can be read and run again.[17, 18] Each event-study result they save includes `eventsUsed`, the exact list of events behind it.

AgriQuant is not a forecasting system. It has no machine-learning model, no training and test data, and no trading strategy. Its ontology is a controlled input layer for an event study: it makes clear what each event means, where it came from, whether it can be analyzed, and which results it fed into. This makes the results explainable in a specific sense. A reviewer can identify the events behind any number, inspect their sources and dates, and repeat the calculation exactly. Because every input can be checked, event-study results built this way can be verified and explained rather than taken on trust.

1.1 Research objective

The objective of this research is to build and evaluate an ontology-backed event list whose rules can be checked, whose results can be traced to their events, and whose analysis can be repeated exactly. The evaluation does not measure investment performance. It looks at whether the records follow the rules, what a full source review changes, whether the results can be regenerated exactly, and how much the results depend on which events are counted.

1.2 Research questions

Four research questions, RQ1 to RQ4, guide the evaluation. **RQ1 (rule conformance)** asks whether every event record carries the required fields, uses only the allowed vocabulary terms,

and agrees with the class its event type implies. **RQ2 (date evidence and eligibility)** asks how often the date an event became public differs from the date it happened, and how requiring a verified public date changes the sample. **RQ3 (repeatable results)** asks whether the same fixed corpus and market files always produce the same events and the same results, without depending on the current date or a live data feed. **RQ4 (sensitivity to event definition)** asks how the sample size and the reported cumulative abnormal return change when the definition of which events count changes.

Each question can be answered with counts and checks. Together they replace broader claims, such as that an ontology improves prediction or that a consistent event list proves cause and effect.

1.3 Contributions

This research makes five contributions. The first is an event model for animal-health and agribusiness financial news, with controlled classes, event types, companies, regulators, species, and directional annotations, and with separate event and public-disclosure dates. The second is an automated integrity check that combines SHACL, exact agreement between the ledger and the Turtle file, and cross-field rules that match each event type to its class, together with queries in SPARQL, the standard query language for RDF,[19] that test the ontology against its seven competency questions. Competency questions are the questions an ontology must be able to answer.[20] The third is a record-by-record review of all 169 original records against primary sources,* in which records that could not be supported were removed or split into separate events, and a public date that could not be confirmed was left blank rather than estimated. The fourth is an event-study implementation that gives the same answer every time: it requires full price history around each event, merges records that fall on the same trading day, groups overlapping event windows, and compares

*The review was carried out with artificial intelligence (AI) assistance, as described in the Preface and Section 3.1.

each result with placebo dates, which are the company's real event dates all moved by the same number of trading days to another part of its price history. The fifth is a fixed analysis setup, made up of a saved configuration, a price file checked by its hash, and results saved as JSON, with tables generated for LaTeX[21] and the exact list of events behind every result.

1.4 Summary of findings

The review of all 169 original records produced 162 final events, and all of them pass the SHACL, ledger, and cross-field checks. The case study analyzes 97 of them: the events of the eight animal-health stocks that were made public from 2021 through 2025. Table 4.1 shows each step from the 169 records to the 97 analyzed events. Each event is compared with its own company's stock, an approach called an own-firm analysis.

The financial result is exploratory. The cumulative abnormal return (CAR) adds up a stock's returns beyond what a market model predicts over a window of days.[1] No company shows a confirmed effect. Zoetis (ZTS) shows one unusual pattern, a mean CAR of -2.97 percentage points from the event day through the tenth trading day after it, but the study makes 40 such comparisons, so this is a lead for a separate test rather than a finding (Section 6.8). The review also changed the sample: 48 of the 97 analyzed events, about half, are analyzed on a different date than the original record gave. Rerun on the original records, the same code gives different results, and ZTS is no longer unusual (Section 6.9). The central lesson of this research is that the review of the event list decides which events and dates enter an event study, and that a reproducible pipeline makes those decisions visible and checkable without turning an exploratory result into a confirmed finding.

1.5 Organization

Chapter 2 reviews event-study methods, ways of representing events as data, forecasting research, and ontology engineering. Chapter 3 describes the research methodology and the

AgriQuant event model and its rules. Chapter 4 explains how the event list was built and reviewed. Chapter 5 describes the software and how it keeps results reproducible. Chapter 6 sets out the event-study method, checks the event list against its rules and competency questions, and reports the case-study results. Chapter 7 answers the research questions, discusses limitations, and concludes. The appendices give the reproduction steps, the field-by-field data contract, the safeguards used against weak practices, and the competency-question queries.

Chapter 2

Background and related work

2.1 Event-study methodology

An event study compares a stock's actual returns around an event with the returns a model predicts. MacKinlay gives a standard overview of the method, Brown and Warner study how it behaves with daily returns, and Campbell, Lo, and MacKinlay place it in a wider econometric setting.[1–3]

The market-model equation is only one part of a sound event study. The event must have the right date. The estimation and event windows must be chosen before anyone looks at the results. Other news released at the same time can blur what caused a price move. Returns can also be correlated across companies and over time, which makes simple standard errors too small. Kolari and Pynnönen develop tests that allow for correlation across companies.[22] Cameron and Miller show that cluster-robust inference becomes unreliable when there are few independent clusters; depending on the design, even 20 to 50 clusters can count as few.[23]

These problems are larger in this research because its event list is manually curated from regulator pages, company news releases, wire services, and publications, not taken from a standardized, pre-cleaned database. The same news can enter the list twice, and

sources can disagree on its date; the source review in Chapter 4 settles both. One company's announcements can also fall within a few weeks of each other, so AgriQuant counts each company trading day once, groups a company's events whose windows overlap, and does not report the usual confidence interval or t-test when a company has fewer than 20 such groups. Finally, as noted above, the returns of companies in the same industry are correlated, so averages across companies are reported only as descriptions.

Agribusiness event studies offer a close example. Golub, Wilson, and Featherstone measured how the StarLink corn food-safety events affected agribusiness stock returns and risk.[24] Their work shows why each domain event needs an exact date before it can be matched to market outcomes. Gondell's thesis on agriculture hedge funds, advised by Featherstone, the agricultural-finance expert for this research, studies agricultural-finance performance and trading strategies.[25] It provides agricultural-finance context but answers a different kind of question. A claim about a fund or a trading strategy needs performance and risk evidence, which AgriQuant does not try to provide.

2.2 Representing events as data

RDF, OWL, and Turtle are standards for storing data as a graph, describing what the data means, and writing it as text.[7, 8, 10] Gruber defines an ontology as an explicit specification of a conceptualization.[9] AgriQuant applies that idea to a narrow scope: it writes down only the concepts and relationships needed to define one event-study record.

Representing events in RDF is well established. Linking Open Descriptions of Events (LODE) is a lightweight vocabulary that links event descriptions to their participants, places, and times.[26] The Simple Event Model (SEM) describes events, actors, places, times, and roles, and deliberately commits to little so that it can work across domains.[27] EventKG builds on SEM to combine more than 690,000 multilingual events and millions of time relations from many sources, while keeping one representation and its provenance.[28]

The Financial Industry Business Ontology (FIBO) is a broad ontology for the financial industry.[29] AgriQuant does not replace FIBO, LODE, SEM, or EventKG. It is a narrow, domain-specific event model built for one workflow. What sets it apart is that it decides which events can be analyzed from their source dates, restricts the event types allowed in each class, returns the exact events behind each result, and produces a fixed event-study output.

SHACL matters because OWL on its own does not reject incomplete data. OWL assumes an open world: a missing value is simply unknown, not wrong. SHACL defines shapes that a data graph must satisfy, so a missing required value becomes an error.[11] AgriQuant uses SHACL for required counts, data types, allowed values, and class-specific event types, and adds a few rules in code where comparing two fields directly is clearer.

Table 2.1 compares the purposes and data controls of these approaches, together with validated comma-separated values (CSV) or JSON files.[30] It is a design comparison, not a performance benchmark.

Table 2.1: *Purposes and data controls of representative event-data approaches*

Representation	Primary scope	Validation and provenance	Relation to AgriQuant
CSV or JSON with validation	Tabular or document data	Schemas, identifiers, constraints, and source metadata can be added	Event definitions and analysis rules are specified by each project
LODE [26]	Linked event descriptions	Reusable relationships among events, participants, places, and times	Provides general event terms for use across domains
SEM [27]	Events, actors, places, times, and roles	Extensible typing and relationships across vocabularies	Provides a reusable model for event representation
EventKG [28]	Large multilingual event knowledge graph	Source integration, temporal relations, and provenance	Integrates events from many knowledge sources
AgriQuant	Animal-health and agribusiness financial-event corpus	SHACL, cross-field checks, source review, and date eligibility	Links reviewed event records to fixed-input event-study results

2.3 Forecasting and machine learning

Research on stock forecasting judges models by how well they predict data they have not seen. Surveys by Rouf et al. and Kumbure et al. summarize the machine-learning methods, data types, and evaluation practices used, and Patel et al. show a typical design for predicting stock and index movement.[31–33]

Work related to agriculture includes neural-network prediction of agriculture and mining stocks, long short-term memory models for commodity prices, models of agricultural prices across space and time, agricultural futures prediction, volatility forecasting, and graph and deep-learning models of futures prices.[34–39] Other studies relate agribusiness stocks to commodity prices, apply decision-tree optimization to agricultural processing stocks, and use topics or news sentiment.[40–43]

These studies show why well-structured inputs can matter, but they do not validate AgriQuant. A forecasting study needs a defined outcome and horizon, protection against information leaking from the future, separate training, validation, and test data, benchmark models, and out-of-sample error. AgriQuant has none of these. Predicting financial events with outside knowledge and graph neural networks is also a different task.[44] AgriQuant is evaluated as infrastructure for building and studying an event list.

2.4 Ontology engineering and reproducible research

The Knowledge Acquisition and Representation Methodology (KNARM) treats knowledge acquisition, expert review, competency questions, and design choices as explicit steps in building an ontology.[12, 13] Related KONCORDANT Lab work applies ontologies and knowledge graphs to grain transportation, food systems, ontology learning, and environmental decision support.[5, 45–48] AgriQuant brings the same commitment to clear, reviewable representations to financial events.

The FAIR principles ask that data be findable, accessible, interoperable, and reusable.[49]

AgriQuant provides stable identifiers, RDF files, source links, and generated outputs. Section 3.5 discusses what long-term publication of the ontology still requires.

Chapter 3

Methodology and ontology design

3.1 Research methodology

This research followed the Knowledge Acquisition and Representation Methodology (KNARM), which builds an ontology or knowledge graph around a concrete use case and keeps domain experts involved throughout.[12, 13] KNARM has nine steps.[5] Table 3.1 shows what this research did at each step and where it is described. Step 5 was not held as a separate interview and step 6 was covered informally; the table says how.

The work began with discussions with Dr. Allen M. Featherstone, the agricultural-finance expert for this research. These discussions identified the kinds of events to collect and what the analysis needed. Candidate announcements were then collected from public sources, with large language model (LLM) assistance used to find an initial list. Reading these announcements showed which details every event needs: the company, the kind of event, its date, its source, and, for regulatory events, the regulator and the target species. It also showed that the date an event happens often differs from the date it is first made public, which led to keeping two dates for every event. These details became the event model described in the rest of this chapter.

Every collected record was then checked against primary sources, with AI assistance from

Codex, OpenAI's coding assistant, and each decision was stored in the verification ledger (Chapter 4). A compiler builds the Turtle knowledge graph from the reviewed ledger. The validated graph feeds the fixed-input analysis that produces the results reported in this research (Chapters 5 and 6), and a small web application shows the events and the saved results. Figure 3.1 shows how these parts connect.

Table 3.1: *KNARM steps as applied in this research*

KNARM step	What this research did	Where described
1. Sub-language analysis	Collected candidate announcements from regulators, company news releases, wire services, and publications, and examined the details they share.	§3.1; Ch. 4
2. Unstructured interview	Discussions with the agricultural-finance expert identified which events to collect and what the analysis needed.	§3.1
3. Sub-language recycling	Reviewed existing event and finance vocabularies (LODE, SEM, EventKG, FIBO). None ties events to disclosure-date eligibility, so a small domain-specific model was built; alignment with them is future work.	Ch. 2; §7.6
4. Meta-data creation and knowledge modeling	Defined the event classes, event types, properties, and the two dates.	§3.3–3.5
5. Structured interview	No structured interview was held for the competency questions. The researcher wrote the seven questions from the analysis needs identified with Dr. Featherstone and from what a reviewer must check in an event study.	§3.2; §6.6
6. Knowledge-acquisition validation	Covered informally rather than in a formal session: discussions with Dr. Featherstone guided the direction of the model, and the source review tested every collected record against it, splitting records that bundled several events.	§3.1; Ch. 4
7. Database formation	Stored review decisions in the verification ledger (JSON) and final events in the Turtle knowledge graph, which every analysis reads directly.	Ch. 4; §5.3
8. Semi-automated ontology building	LLM assistance suggested the candidate list; a compiler writes the Turtle event records from the reviewed ledger.	§3.1; §4.6
9. Ontology validation and evaluation	SHACL and cross-field checks, a logical-consistency check with the HermiT reasoner [50], and SPARQL queries for the competency questions.	§3.6; §6.5–6.6

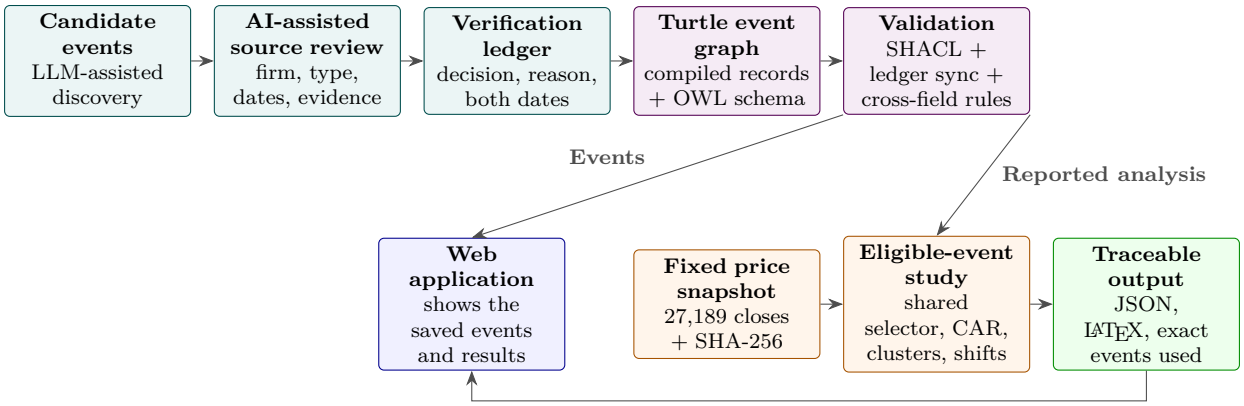


Figure 3.1: *End-to-end AgriQuant workflow. Teal boxes are review steps, violet boxes the knowledge graph, orange boxes the reported analysis, the green box its outputs, and the blue box the web application. AI-assisted source review is recorded in the verification ledger, which is compiled into the Turtle event graph and validated. The validated events feed the fixed-input analysis, and the web application shows the events and the saved results without computing anything itself.*

3.2 Design requirements

The ontology was built to answer the questions a reviewer asks when checking an event study. These competency questions are:

1. Which company does the event belong to?
2. Which event class and controlled event type describe it (for example, class `CorporateEvent` and type `acquisition` when one company buys another)?
3. What date is given for the event itself, and what kind of date is it?
4. Which primary source supports the event, and when was it first made public?
5. Is there a regulator or a target species?
6. Which directional annotation was assigned, and why?
7. Which exact events went into a reported number?

The first six questions describe the event record. The seventh connects it to the analysis. Each question is also written as a SPARQL query, and Chapter 6 reports what the queries

return. The ontology is kept small on purpose. It answers these questions and does not try to model every financial instrument, biological concept, regulatory process, or detail of how markets trade.

3.3 Core classes

The root `ag:Event` class has six subclasses, listed in Table 3.2. The subclasses are disjoint, so an event can belong to only one of them: a record cannot be both an approval and an acquisition. Disjointness alone does not stop one record from bundling two events of the same class, such as two approvals. The source review handles that case by splitting such a record into separate events, as Chapter 4 describes.

Table 3.2: *The six AgriQuant event classes*

Event class	What it covers
DrugApproval	Regulatory approval, marketing authorization, registration, license, or label expansion
PipelineEvent	Development-pipeline milestone or update
ScientificDisclosure	Original study publication or scientific presentation
RegulatoryAction	Warning, fine, enforcement action, emergency-use authorization, regulatory opinion, or other regulatory update
CorporateEvent	Acquisition, joint venture, capacity change, restructuring, production disruption, or other controlled corporate event
ProductLaunch	Commercial launch of a product

Despite its name, DrugApproval holds any regulator's approval or registration of a product, not only veterinary drugs. For example, FMC's 2025 registration of the herbicide active ingredient Dodhylex in South Korea is a DrugApproval.

Company, EventType, Sentiment (the class that holds the directional-annotation values), Species, Registry, and MarketCategory are separate classes or controlled vocabularies. Each event type names the event class it belongs to, so a mismatch can be caught automatically. For example, `regulatory_approval` must be a DrugApproval, and `acquisition` must be a CorporateEvent.

3.4 Required and optional properties

Table 3.3 lists the properties the analysis relies on. A value that points to another resource, such as a company or an event type, is an internationalized resource identifier (IRI), the global name that RDF gives to a resource.[51] Source links are uniform resource locators (URLs), the web addresses of the sources.[52] The counts are enforced by the SHACL shapes and the checks in notebooks 1 and 2; they are not left to OWL inference.

Table 3.3: *Core AgriQuant event properties*

Property	Count	Value kind	Purpose
`hasCompany`	exactly one	Company IRI	Target firm and event-universe selection
`hasEventType`	exactly one	EventType IRI	Controlled semantic subset and expected class
`hasSentiment`	exactly one	controlled IRI	Directional annotation assigned by the researcher
`eventDate`	exactly one	date literal	Date of the decision, transaction, launch, incident, or announcement
`publicDisclosureDate`	zero or one	date literal	Earliest exact contemporaneous public disclosure; analysis date
`analysisEligible`	exactly one	Boolean	True exactly when public disclosure date is present
`dateKind`	exactly one	string	Meaning of the event date
`title`	exactly one	string	Human-readable event identity
`sourceUrl`	exactly one	URL literal	Primary source access path
`sourceTitle`	exactly one	string	Identifies reviewed primary evidence
`sourcePublisher`	exactly one	string	Publisher responsible for the evidence
`annotationRationale`	exactly one	string	Rationale for directional annotation
`derivedFromEventId`	exactly one	string	Original record reviewed for this event
`targetSpecies`	zero or more	Species IRI	Veterinary scope; multivalued where appropriate
`hasRegistry`	zero or one	Registry IRI	Regulator for regulator-backed events

These counts are design choices. Each event has exactly one company, so the own-firm

event study always has one unambiguous stock to compare against. Each event has exactly one event type, which gives every record one controlled classification, and because each type maps to an expected class, a conflict between the two is caught automatically. Each event has exactly one directional annotation, so subsets can be filtered consistently, while the annotation stays marked as the researcher's judgment rather than a fact from the source. The registry is optional because many corporate and commercial events involve no regulator.

Species can have more than one value on purpose. SHACL testing showed that Pradalex is approved for both cattle and swine, so a one-species limit would have rejected a correct record. The lesson is that a rule is not right just because it runs. Constraints still need review by someone who knows the domain.

The schema was inspected in Protégé, an ontology editor.[53] Figure 3.2 shows the schema in OntoGraf, a Protégé plugin that draws an ontology as a graph. The view is centered on `Event` and shows its six subclasses and the classes that its properties link to.

3.5 Identity and authoritative fields

Each final event identifier is generated the same way every time from the reviewed record, and it includes the corrected event date. The `derivedFromEventId` field keeps the original record's identifier, even when that identifier contains an old, wrong date. Identifiers are only keys. Software must never read the date, company, or type out of an identifier.

The current namespace, the common prefix of the ontology's identifiers, points to files in the project repository. This supports inspection and version control, but it is not a full long-term publication plan. A future release should use a stable web domain or an archive that assigns a digital object identifier (DOI), a persistent identifier for published material.[54] It should also support content negotiation, in which the server returns the format a client asks for, such as Turtle for software or a web page for a person.

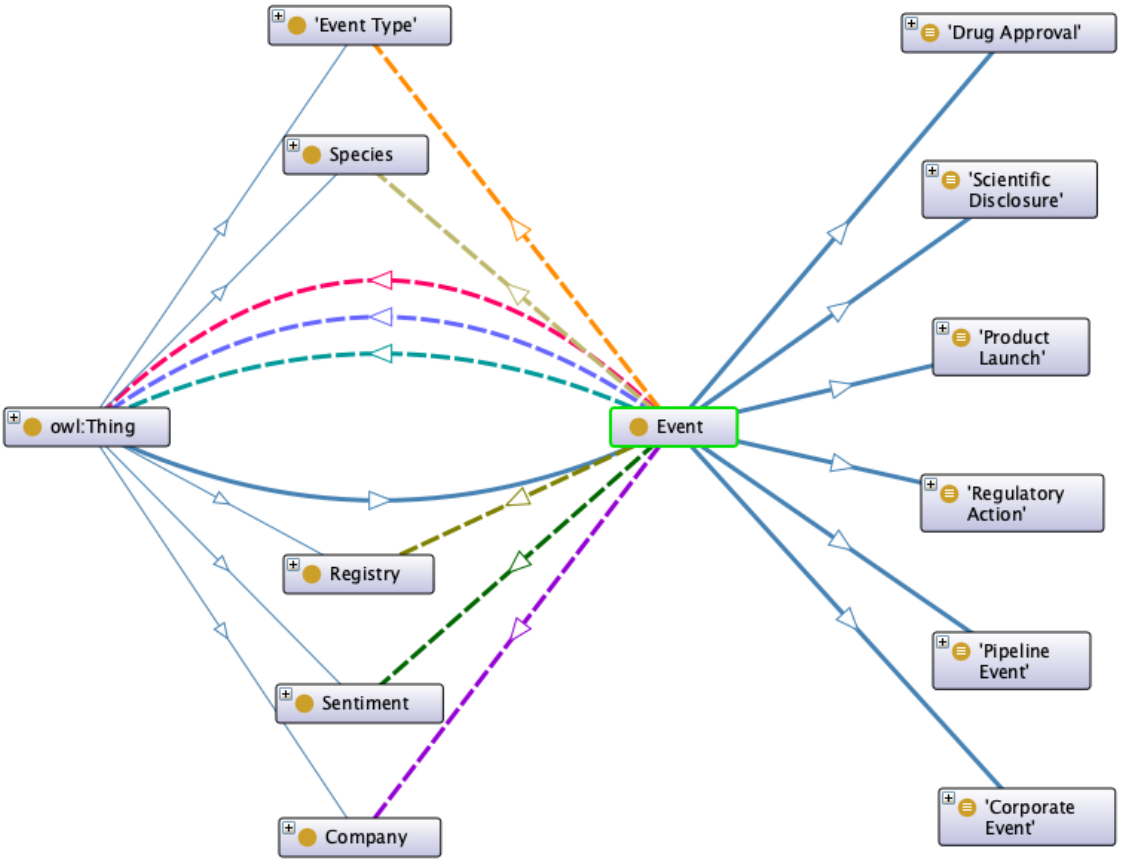


Figure 3.2: *OntoGraf view of the AgriQuant event schema in Protégé Desktop*

3.6 OWL semantics and SHACL validation

OWL defines the concepts: subclasses, event classes that cannot overlap, property domains and ranges, and the mapping from event types to classes. SHACL defines what each stored record must look like. The EventShape applies to all six event classes and checks required counts, data types, allowed values, and URL patterns. Class-specific shapes limit each class to its permitted event types.

Analysis eligibility uses a conditional shape. An eligible event must have exactly one public-disclosure date, and a fact-only event must have none. Both must still have an event date. This stops a regulator's internal decision date from quietly becoming the date the market is assumed to have learned the news.

A separate check in notebook 2 (Section 5.2) merges the schema with the event records using ROBOT, a command-line tool for ontology workflows.[55] It then runs HermiT, a reasoner that tests whether an OWL ontology contains a logical contradiction, over the merged graph.[50] This complements SHACL, which checks shapes but does not apply `rdfs:domain` rules. Such a rule lets a reasoner infer an event's class from a property the event carries. Appendix C describes the domain error this check found. The schema declares `xsd:date` explicitly so that HermiT accepts the dates, and the merged graph passes the HermiT consistency check.

Notebooks 1 and 2, run whenever the ledger changes, add further checks. Every original record must have a final decision, final identifiers must be unique, directional labels must be allowed, and event classes and types must agree. The Turtle file is rebuilt from the ledger on every run, so the two always agree, and excluded originals cannot return. Notebook 1 also prints short coverage summaries. A missing annotation, rationale, source detail, or eligibility value stops the run. Notebook 2 deliberately removes a required annotation from a temporary copy of the data to verify that SHACL detects the missing value.

3.7 Directional annotation

Each event carries a directional annotation that records whether the researcher expected the news to help or hurt the company. The annotation is stored as `ag:hasSentiment` with the values `good`, `bad`, and `neutral`, which the tables label favorable, adverse, and neutral. Each annotation comes with a written rationale. Most rationales are category-level, written per kind of event rather than per record: the 162 events share 28 distinct rationale texts, and the most common text is used by 42 events. The annotations are not facts taken from the source, and they are not the stock's actual return. They are kept because they let each event state its expected direction and let the analysis select subsets by direction (Section 6.11). They are not treated as tested predictors. Most events are labeled favorable, and the labels are closely tied to event type. Chapter 6 reports this limitation.

Chapter 4

Corpus construction and source review

4.1 Corpus scope

The reviewed corpus covers 18 publicly traded companies. Eight are the animal-health companies of the case study (Table 6.1): Zoetis, Elanco, Phibro, and Merck in veterinary medicines, IDEXX and Neogen in veterinary and food-safety diagnostics, and Charles River and Inotiv in research animals and the testing services built on them. The other ten are agribusiness companies: Deere, AGCO, and CNH Industrial in farm equipment, Nutrien, Mosaic, and CF Industries in fertilizer, Corteva in seeds, FMC in crop chemicals, and Archer-Daniels-Midland and Bunge in grain trading and processing. Its events fall into the six event classes (Table 4.2), with event dates from 2013 through 2025. Every final event has a source URL, source title, source publisher, and directional annotation. A registry is recorded when a regulator or formal registry is involved.

The two groups play different parts. The animal-health companies hold most of the events and supply the case study. The agribusiness companies hold three events each. They were collected to ask whether animal-health events also move agribusiness stocks, a question

that needs many more events and is left for future work (Section 7.6). In this research they show that the event model and the source review apply outside animal health.

The corpus does not list every announcement these companies made. The sources that were available shaped the original sample. The review settles what happened to each original record. It does not make the sample representative of everything these firms disclosed.

4.2 Why a source link is not enough

Having a source URL for every record does not prove that the timing is right. A product page can show that an event happened without saying when it happened or when investors could have known. Product and regulator pages often display a later update date. Before the review, 56 of the 169 original records gave only the month of an event, stored as the first day of that month as if it were exact. Other records repeated the same approval, described events after the fact, or bundled several legally distinct events into one row.

The review therefore asks four separate questions. The first is whether primary evidence supports what the event is, which company it belongs to, and how it is classified. The second is what exact date describes the decision, transaction, incident, launch, or announcement itself. The third is the earliest exact date on which primary evidence shows the news was made public. The fourth is whether the record describes one event, distinct from every other event kept.

Asking these questions separately stops a hidden regulatory decision date, a later page-update date, or a month placeholder from becoming the event-study date.

4.3 Review protocol and primary sources

The review uses fixed fields, decisions, and standards of evidence. Each fact is checked against the most appropriate primary source: official records for regulatory or legal decisions, dated company announcements for corporate events, and original research papers for scientific

findings. Undated marketing pages, search-result snippets, later secondary reporting, and plain product pages are not enough on their own to set an exact public-disclosure date.

The researcher completed the record-by-record review on July 12, 2026. The source-evidence file, `source-evidence.json`, lists source references for all 169 original records. It covers 137 source webpages (98 fetched automatically, 39 blocked or failed) and includes 14 application records from Animal Drugs @ FDA, the database of approved animal drugs kept by the U.S. Food and Drug Administration (FDA), and 9 FDA Freedom of Information (FOI) summaries, the documents the FDA publishes when it approves an animal drug. The file `event-verification-ledger.json` records, for every original record, the final review decision, source details, the event date, any public-disclosure date, the classification, the annotation rationale, how it was reviewed, and when. These recorded review decisions form the basis for the final corpus; neither file replaces an independent second review.

Every original record ends with one of two decisions: `verified` or `excluded`. Nothing is left unresolved. When the evidence cannot support a single, exactly dated event, the record is excluded rather than given a made-up date. A record that combined several events can be replaced by separate records, one per event. A duplicate is simply excluded.

4.4 Review outcome

All 169 original records received a decision: 155 were kept and 14 were removed. Five of the removed records repeated an event already in the list. Four used the date on which the European Medicines Agency published a web page as if it were a regulatory event. One described an inspection that ran for months rather than a single dated event. The last four each combined two events in one record or relied on a secondary news report, and they were replaced by 7 records that each hold one event confirmed by a primary source. The 155 kept records and the 7 replacements make the 162 final events. The review also corrected 78 event dates. Table 4.1 follows the records from the review to the case study.

Table 4.1: *From original records to analyzed events*

Step	Count
Original records reviewed	169
Removed by the review	−14
Single-event replacements added	+7
Final events in the corpus	162
Kept as facts only, with no confirmed public date	−2
Events that can enter an analysis	160
Events of the ten agribusiness companies	−30
Animal-health events made public before 2021	−33
Events analyzed in the case study	97

Table 4.2 shows the final events by class and whether each can enter the analysis.

Table 4.2: *Final events and disclosure-date eligibility*

Event class	Final events	Analysis-ready	Fact-only
DrugApproval	60	58	2
PipelineEvent	5	5	0
ScientificDisclosure	2	2	0
RegulatoryAction	20	20	0
CorporateEvent	52	52	0
ProductLaunch	23	23	0
Total	162	160	2

The two fact-only events have exact regulator decision dates but no confirmed public announcement from the time: the 2017 Banamine Transdermal approval and the 2022 Stafac/Maxiban feed-combination approval. They stay in the corpus as facts but cannot be used in the event study.

Figure 4.1 shows one event and its connections in Protégé: the FDA approval of Zoetis's Librela for osteoarthritis pain in dogs, approved and announced on May 5, 2023. The event is linked to its ontology class, its company, its regulator, its target species, its event type, and its directional annotation. The full ontology holds all 162 final events listed in Table 4.2.

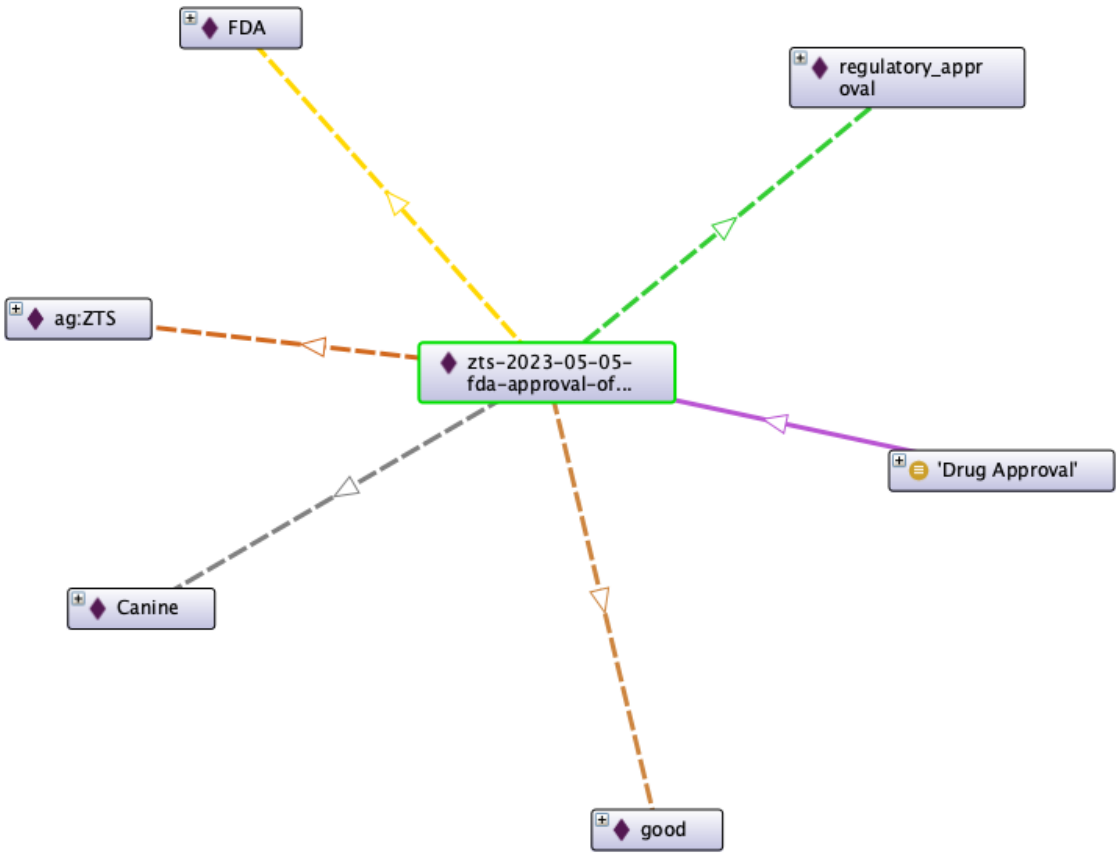


Figure 4.1: *The May 5, 2023 FDA approval of Librela and its connections in Protégé OntoGraf*

4.5 Event dates and disclosure dates

The ontology keeps `eventDate` and `publicDisclosureDate` apart. The first says when something happened. The second says the earliest exact date on which reviewed public evidence announced it. For 49 final events the two dates differ. Examples include regulatory decisions announced a few days later and an acquisition whose legal closing date was corrected in a later company filing. The largest gap in the case study belongs to a Simparica Trio label expansion for the Asian longhorned tick: the FDA approved it on November 6, 2024, but Zoetis first announced it on January 14, 2025, 69 days later, and the case study uses January 14.

Keeping the dates apart also lets the corpus hold an event as a plain fact when its public date is unknown. `analysisEligible` is true exactly when `publicDisclosureDate` is present. SHACL enforces this rule, and notebook 1 checks it again. The event study lines up only the public-disclosure date with the trading calendar.

4.6 Building the Turtle file

The compiler in notebook 1 (Section 5.2) reads only the ledger and writes `ontology/events-data.ttl`. It stops on missing decisions, invalid dates, unknown ontology mappings, provenance links that do not point to an original record, or clashing final identifiers. The file is built only from kept records and their replacements, so an excluded original cannot come back unless an explicit replacement record stands in for it. Every run reports whether the rebuilt file differs from the tracked one, and on the tracked ledger the two match byte for byte.

Notebook 1 also checks that the ledger is complete and that source and annotation details are present, and notebook 2 checks that SHACL passes and that classes and types agree. The expected result is 162 final events, 160 analysis-ready events, 2 fact-only events, and zero errors.

4.7 What the review changed for the analysis

All 97 analyzed events (Table 4.1) have complete price windows. They fall on 93 distinct company trading days, and all eight companies contribute. The review changed this sample in two ways. First, before the review, the original records marked 45 of the 169 dates as checked against a source and the other 124 as still to be checked. Holding back the 124 would have cost most of the case study: 67 of the 97 analyzed events come from them, including every analyzed event of IDEXX Laboratories (IDXX) and Charles River Laboratories (CRL). The 45 checked dates were not safe either, because the review changed the event date of 13 of them. A date marked as checked was therefore no guarantee, which is why every record was reviewed again rather than only the unchecked ones. Second, 48 of the 97 analyzed events are analyzed on a different date than the original record gave.

These counts show that the review changed both which events enter the event study and the dates on which they are analyzed. Section 6.9 reruns the case study without the review to show how much that changes the results. The financial estimates are still exploratory, because a full source review does not solve overlapping news, the small number of non-overlapping clusters, multiple testing, or the way the corpus was first assembled.

Chapter 5

System and reproducibility architecture

5.1 Architecture overview

AgriQuant is a research pipeline with a small web application on top. Figure 3.1 groups its parts by color: review steps (teal), the knowledge graph (violet), the reported analysis (orange), its outputs (green), and the web application (blue). The knowledge graph holds the OWL schema, the Turtle event records, and the SHACL shapes that check them. The pipeline is three Jupyter notebooks written in Python (Section 5.2). They build the event file from the verification ledger, check the knowledge graph, and run the event study on a fixed copy of market prices. The web application shows the events and the results that the notebooks saved, and it computes nothing itself (Section 5.8).

5.2 Notebooks

Each notebook runs from top to bottom as a sequence of short steps, and every step has a sentence above it that says what it does. A step that checks something stops the notebook when the check fails, so a notebook that runs to the end has passed all of its checks. Table 5.1

lists what each notebook does and which files it writes.

Table 5.1: *The three AgriQuant notebooks*

Notebook	What it does	Files it writes
1. Review ledger	Checks that the review decisions are complete and consistent, and builds the final events from them (Chapter 4).	`ontology/events-data.ttl`; the event data for the web application
2. Ontology checks	Runs SHACL, the type and class rule, the competency questions, and the HermiT consistency check, then shows that the checks catch deliberate errors (Sections 3.6 and 6.6).	None
3. Event study	Works through one event step by step, then runs the case study, the placebo check, the sensitivity subsets, and the comparison without the review (Chapter 6).	The results in `results/`; the study data for the web application

A notebook writes a file only when the file's contents change, and it reports for every file whether it was unchanged or updated. On the tracked inputs every file is reported unchanged, which shows that the saved outputs are exactly what a fresh run produces.

5.3 Reading events from the graph

The notebooks read the final events straight from the knowledge graph. They load the ontology schema and the event records with rdflib, a Python library for reading and querying RDF, and turn each event into one flat record: its identifier, the original record it came from, its class, company, event date, public-disclosure date, eligibility, date kind, title, type, registry, species, directional annotation and rationale, and full source details. Company tickers and names come from the named company nodes. There is no separate database copy of the events, so any addition, edit, or deletion in the final corpus reaches the next run of every notebook.

5.4 Shared event selection

One function in notebook 3 decides which events enter the case study and its sensitivity checks. For each stock, it chooses events in two ways: by event source (approval-class events, other events, or both) and by directional annotation (favorable, adverse, neutral, or all). Events are always matched to the stock's own company, because the ontology has no relation that says why one company's news should affect another company's stock.

The selection does not remove duplicates during analysis. The curated Turtle file lists each real event once, so duplicates are handled during curation and validation rather than by a hidden switch in the analysis. If two different source records fall on the same company trading day, the event study treats them as one statistical unit but keeps both identifiers.

The case study and every sensitivity subset use this one function, and each saved result includes `eventsUsed`, the exact list of events behind it. Prices come only from the fixed snapshot (Section 5.5).

5.5 Market snapshot

The analysis uses a fixed dataset of daily closing prices from Stooq, covering June 1, 2020 to February 9, 2026, for 18 stocks and SPY.[15] SPY tracks the S&P 500 and serves as the market benchmark. The dataset contains 27,189 observations. All notebooks use this saved dataset, so later changes to the provider's data do not affect the reported results.

5.6 Analysis identity

Each analysis gets an identifier built by hashing the stock, the selected event identifiers and their public-disclosure dates, the full stock price series, and the full benchmark price series. Hashing only the date range would not be enough, because a provider could revise prices inside the same dates. The identifier changes whenever the events or the prices change.

The random seed comes from this identifier. A seed is the starting value of a random-number generator, so the same seed always gives the same random draws. Here the seed fixes the order in which circular shifts are tried (Section 6.4). Selections that lead to the same events and prices give the same seed, so the same sample always gets the same placebo results. The order matters only when the shift limit is smaller than the number of possible shifts. In this case study every possible shift is tried, so the seed changes no reported value, and it serves as the analysis identifier.

5.7 Generated outputs

Notebook 3 writes three kinds of output: a JSON file[14] with the configuration, price manifest, corpus review summary, exact events, analysis identifiers, totals, exclusions, sensitivity summaries, and the comparison without the review; LaTeX table fragments[21] for own-firm CAR, placebo diagnostics, subset sensitivity, and the comparison without the review; and LaTeX macros for analysis dates, counts, snapshot identity, and the placebo iteration limit.

Because each file is rewritten only when its contents change, a rerun on the same inputs must report every file as unchanged, which means the files match byte for byte. This catches mistakes such as using today's date, querying live prices, iterating in an unstable order, or using an unstable random seed.

5.8 Web application

The web application shows what the notebooks saved: the final events from notebook 1 and the case-study results from notebook 3. It is written in TypeScript, a typed form of JavaScript, with React, a JavaScript library for building user interfaces, and it has four views. The overview shows the counts from the 169 reviewed records to the 162 final events and the 97 analyzed events, one example event with both of its dates and its source, each event class's share of the events, and the archived price file with its SHA-256 hash. The

event browser lists every final event with its company, event date, public-disclosure date, eligibility, and source link, and each record shows its ontology class, event type, registry, original record, and the reason for its directional annotation. The study view shows the case study one company at a time. A single row of the eight stocks, each with its CAR$[0, +10]$, picks the company. Below it the view shows that company's number of events, distinct event days, and overlap clusters, its CAR for each window against the circular-shift results, its daily abnormal returns, the same result without the review, the exact source events used, and the events left out with the reason. The files view lists every research file, from the notebooks and the Turtle files to the ledger, the price snapshot, the results, and the thesis document, and opens each one read-only. Throughout the application, the saved values link to the file they come from, and each event links to its own entry in the Turtle event file and in the verification ledger. The ontology schema, and the graph of links around each event, are viewed in Protégé (Figures 3.2 and 4.1).

The application has no setting that changes a result, so it cannot show a number the notebooks did not produce. Where inference is withheld, the study view says so and why, for example "Too few for a margin of error (needs 20)" beside a count of 7 clusters, rather than showing the empty values as zeros.

Chapter 6

Event-study design and results

6.1 Evaluation protocol

The evaluation has two parts. The first checks whether the reviewed corpus follows the data contract, whether the ontology answers the competency questions it was designed for, and whether the saved outputs can be regenerated exactly. The second uses the events with confirmed public dates in an own-firm event study. The first part evaluates the system, and the second shows what happens when it is used. A return estimate on its own says nothing about whether the ontology meets its requirements.

The case study covers the eight stocks listed in Table 6.1 and analyzes events publicly disclosed from 2021-01-01 through 2025-12-31. The interval applies to the public-disclosure date, not the event date. For example, Charles River Laboratories' acquisition of Distributed Bio was completed on December 31, 2020 and announced on January 4, 2021, so it is analyzed. An event enters a company's sample only if the ontology links it to that company and the review found an exact public-disclosure date.

The eight companies differ in how much of their business their animal-health events describe. Zoetis, Elanco, and Phibro sell mainly animal-health products. Merck is a different case: its Animal Health segment had sales of about $5.9 billion in 2024, about 9% of the

Table 6.1: *The eight stocks in the animal-health case study*

Ticker	Company
ZTS	Zoetis Inc.
ELAN	Elanco Animal Health
PAHC	Phibro Animal Health Corporation
MRK	Merck & Co., Inc.
IDXX	IDEXX Laboratories, Inc.
NEOG	Neogen Corporation
CRL	Charles River Laboratories
NOTV	Inotiv, Inc.

company's $64.2 billion total, and the rest came mostly from human medicines.[56] Charles River and Inotiv sell research animals and testing services to drug developers rather than veterinary products. For Merck in particular, an animal-health event is a small part of the news that moves the stock, so a small or unusual Merck CAR says little about whether the event mattered to the business. The own-firm design keeps each company's events matched to its own stock, but the results are not comparable across companies as measures of the same exposure.

6.2 Return and market model

Let $P_{i,t}$ be the provider's closing price for stock i on trading day t. The daily log return is

$$R_{i,t} = \log\left(\frac{P_{i,t}}{P_{i,t-1}}\right). \tag{6.1}$$

SPY is the market benchmark, and $R_{m,t}$ is its daily log return. For each event, ordinary least squares fits the market model

$$R_{i,t} = \alpha_i + \beta_i R_{m,t} + \epsilon_{i,t} \tag{6.2}$$

using exactly 120 days of matching stock and benchmark returns, from trading day -130 through day -11 relative to the event. The event window runs from day -10 to day $+20$.

An event is dropped if either period is incomplete, so a short price history near the edge of the data is never treated as comparable.

Each public-disclosure date is moved to the first day on or after it when both the stock and SPY traded. This handles weekends and holidays. It cannot tell whether a same-day announcement came before or after the market closed, because the ontology does not yet record the time of day or the time zone.

Using the fitted coefficients, the abnormal return is

$$AR_{i,t} = R_{i,t} - (\widehat{\alpha}_i + \widehat{\beta}_i R_{m,t}), \tag{6.3}$$

and the cumulative abnormal return (CAR) between days τ_1 and τ_2 is

$$CAR_i(\tau_1, \tau_2) = \sum_{t=\tau_1}^{\tau_2} AR_{i,t}. \tag{6.4}$$

Because the returns are log returns, a CAR is a sum of log abnormal returns. The tables report it multiplied by 100 and call the unit percentage points. The two agree closely for small moves and drift apart for very large ones, so a CAR can fall below -100. The output records CAR for $[-1, +1]$, $[-5, +5]$, $[0, +10]$, $[0, +20]$, and $[-10, +20]$. The tables focus on the short window around the announcement, $[-1, +1]$, and the window from the event day through day $+10$, $[0, +10]$. These windows were fixed in the event-study code before any company's result was examined. The window lengths, like the 120-day estimation period, are design settings, the event-study counterpart of hyperparameters in machine learning: the researcher chooses them, they are not estimated from the data, and they must not be tuned on the results. Starting ten trading days before the event allows for news that is expected or leaks early, and ending twenty trading days after it allows for slower reactions. The same window also decides which events overlap (Section 6.3).

6.3 Analysis units and dependence

A row in the corpus and a unit in the statistics are not always the same thing. When several source records for one company fall on the same trading day, the analysis merges them into one event-day unit and keeps every event identifier for tracing. This stops several descriptions of the same day from inflating the sample.

A company's event windows can also overlap. Because each window runs from day -10 to day $+20$, two events of the same company share trading days when they fall within 30 trading days of each other. The analysis groups chains of such overlapping windows into clusters, so one cluster can span more than 30 trading days. For example, the 21 ZTS events form seven clusters. The clusters change only the measure of uncertainty: every event day still counts once in the mean CAR. With enough clusters, uncertainty could be estimated with cluster-robust standard errors or with a cluster bootstrap, which repeatedly resamples whole clusters to see how much the mean would vary.[57] The analysis reports no confidence interval or t-test, however, when fewer than 20 clusters remain. This cutoff is cautious on purpose, because cluster-robust inference is known to be unreliable with few clusters.[23] All eight companies fall below it, so these fields are left empty rather than shown as zeros.

6.4 Circular-shift placebo check

The placebo check asks whether a company's observed mean CAR is unusual compared with what the same spacing of events gives at other points in that company's price history. If the observed event positions are $e_1, \ldots, e_k$, a shift s moves them to

$$e_j^* = (e_j + s) \bmod T, \tag{6.5}$$

where T is the length of the usable trading calendar. The shift is never zero. A shift is rejected if any moved date lands on a real event date or cannot support the same complete

estimation and event windows. Each accepted shift is analyzed with the same market model.

The implementation steps through allowed shifts up to the configured limit of 2000. With $T = 1{,}280$ usable trading days, each company has 1,279 possible nonzero shifts, fewer than this limit, so every shift is tried and the order in which they are tried does not affect the result. Between 897 (MRK) and 1,267 (PAHC) shifts are accepted, depending on the company. For ZTS, 368 of the 1,279 shifts would put an event on a real ZTS event day, which leaves 911. A shift of 640 trading days, for example, moves the first ZTS event day, February 22, 2021, to September 7, 2023, and moves every other ZTS event day by the same amount. The output reports the average placebo mean CAR$[0, +10]$ and the range that holds the middle 95% of placebo results. The two-sided diagnostic is

$$p_{\text{shift}} = \frac{1 + \#\{|\overline{CAR}^{*}| \geq |\overline{CAR}|\}}{1 + B}, \tag{6.6}$$

where B is the number of accepted shifts. Adding one to the top and bottom counts the real event timing as one of the timings compared. This is the standard correction for such checks, and it means the value can never be exactly zero.[58] The check compares the real timing with other possible timings. It is not a causal test, and it does not account for all other news at the time.

6.5 RQ1: rule conformance

The final files contain the 162 events of Table 4.2. Notebooks 1 and 2 report zero SHACL violations and no failures of the other rules. Every event has one identifier, a link to its original record, a company, an event type that matches its class, an event date, an eligibility flag, a date kind, a title, a directional annotation with its rationale, and full source details. Eligible events have exactly one public-disclosure date, and fact-only events have none.

This answers RQ1 only for the rules that were written down. A graph can pass every rule and still miss a real event or contain a reasonable but debatable curation decision. Passing

the checks shows that the data is internally consistent, not that it is complete.

6.6 Competency questions

Chapter 3 lists seven competency questions that the ontology was designed to answer. Writing such questions down and testing them is a standard way to check whether an ontology does its job.[12, 20] Each question is written as a SPARQL query over the merged schema and event files (Appendix D). Notebook 2 runs all seven queries with rdflib and checks that each one returns the expected number of answers. Table 6.2 summarizes what the queries return, numbered by competency question (CQ); notebook 2 derives every count in it from the query answers.

Table 6.2: *What each competency-question query returns*

CQ	Question	Answer from the query
1	Which company does the event belong to?	One company for each of the 162 events, across 18 companies
2	Which event class and controlled event type describe it (for example, class `CorporateEvent` and type `acquisition` when one company buys another)?	All 162 events, each with a type that matches its class (6 classes, 23 types)
3	What date is given for the event itself, and what kind of date is it?	A date and a date kind for all 162 events
4	Which primary source supports the event, and when was it first made public?	A source URL, title, and publisher for all 162 events; 160 also have a public-disclosure date
5	Is there a regulator or a target species?	85 events have a regulator and 96 have at least one species; 58 have neither
6	Which directional annotation was assigned, and why?	A label and rationale for all 162 events: 145 favorable, 12 adverse, and 5 neutral
7	Which exact events went into a reported number?	97 events, exactly the events used in the eight case-study results

Every query returns an answer for every event it should. Query 2 also checks that no event type disagrees with its class: it joins each event to the class its type expects, so a mismatch would drop the event from the answer. The counts agree with the corpus summary that

notebook 3 computes from the same files through separate code. Query 6 returns a rationale for every event, but most rationales are category-level, written per kind of event rather than per record (Section 3.7).

Query 7 links the ontology to the reported results. Using only the graph, the eight tickers, and the analysis period, it rebuilds the case-study sample, and the 97 events it returns are exactly the events listed in the saved results. A reader can therefore trace the case-study results back to their events through the ontology alone, without running the event study. To make sure these checks can fail, notebook 2 makes deliberate errors in a copy of the data. Giving one approval an acquisition type drops it from the answer to query 2, and moving one public date outside the analysis period drops it from the answer to query 7.

These results show that the ontology can answer the questions it was built for. They do not show that these are the right questions, or that the answers help analysts in practice. Judging that needs domain experts, as Chapter 7 discusses.

6.7 RQ2: date evidence and eligibility

Every original record received a final decision (Table 4.1), and the review corrected 78 event dates. Of the 162 final events, 160 have exact public-disclosure dates, and 2 are kept as facts without an analysis date. For 49 events, the event date and the disclosure date differ.

Within the fixed period and the eight stocks, 97 events with confirmed public dates are analyzed. For 36 of them, the analysis uses a public-disclosure date that differs from the event date, and 48 are analyzed on a different date than the original record gave. The eligibility and date rules therefore change both the size of the sample and the timing of its events, and Section 6.9 shows that they also change the results.

6.8 Case-study results

Table 6.3 reports descriptive CAR for each company. R is the number of events analyzed, D is the number of distinct event days, and G is the number of non-overlapping clusters, where each cluster is a group of overlapping event windows. No company reaches 20 clusters.

Table 6.3: *Descriptive own-firm CAR in percentage points*

Ticker	R	D	G	CAR$[-1,+1]$	CAR$[0,+10]$
ZTS	21	21	7	-0.89	-2.97
ELAN	18	17	8	+0.24	-2.58
PAHC	5	4	3	-0.39	-5.47
MRK	23	22	12	-0.61	-1.14
IDXX	5	5	5	-1.10	-0.06
NEOG	6	6	6	+0.51	-1.42
CRL	8	8	6	-0.28	+2.83
NOTV	11	10	6	-8.31	-9.70

NOTV has the most negative mean CAR$[0,+10]$, -9.70 percentage points, from ten event days in six clusters. ZTS is at -2.97 percentage points from 21 event days in seven clusters. These numbers mean little without the sample sizes, the cluster structure, and the placebo context.

Table 6.4 gives that placebo context. The interval shown is the range holding the middle 95% of placebo results. It is not a confidence interval for the observed CAR.

Table 6.4: *Circular-shift diagnostics for mean CAR*$[0,+10]$ *in percentage points*

Ticker	G	Shift mean	Shift 95% interval	p_{shift}
ZTS	7	-0.10	[-2.53, +2.23]	0.012
ELAN	8	+0.15	[-3.65, +4.07]	0.230
PAHC	3	+0.39	[-7.82, +9.85]	0.226
MRK	12	+0.18	[-1.74, +2.21]	0.281
IDXX	5	-0.02	[-4.63, +4.89]	0.987
NEOG	6	+0.74	[-6.94, +8.78]	0.727
CRL	6	+0.16	[-6.53, +6.01]	0.315
NOTV	6	-1.05	[-15.64, +11.82]	0.176

The ZTS mean CAR$[0,+10]$ is unusual compared with its placebo distribution, with

$p_{\text{shift}} = 0.012$. The other companies range from 0.176 to 0.987. ZTS has only seven non-overlapping clusters, every company is below the 20-cluster cutoff, and the placebo values are not adjusted for testing several companies and windows. Table 6.5 lists p_{shift} for every company and every CAR window, which makes 40 comparisons. Three values fall below 0.05: ZTS for CAR$[0, +10]$ (0.012) and CAR$[-5, +5]$ (0.011), and NOTV for CAR$[-1, +1]$ (0.031). Each comparison has about a 1-in-20 chance of falling below 0.05 even when the events have no effect, so chance alone would produce about two such values (0.05×40). The ZTS result is therefore a lead, not a finding, and the study makes no confirmed claim about any company.

Table 6.5: *Placebo p_{shift} for every company and CAR window. Values below 0.05 are in bold.*

Ticker	$[-1, +1]$	$[-5, +5]$	$[0, +10]$	$[0, +20]$	$[-10, +20]$
ZTS	0.107	**0.011**	**0.012**	0.115	0.127
ELAN	0.794	0.468	0.230	0.280	0.163
PAHC	0.840	0.356	0.226	0.215	0.298
MRK	0.254	0.379	0.281	0.839	0.898
IDXX	0.363	0.593	0.987	0.840	0.583
NEOG	0.790	0.878	0.727	0.580	0.802
CRL	0.806	0.853	0.315	0.757	0.950
NOTV	**0.031**	0.766	0.176	0.204	0.213

Because every result lists its events, a reader can also open the ZTS result itself. Of its 21 events, 18 carry a favorable annotation, mostly approvals and label expansions, and 3 carry an adverse one, the 2024–2025 warnings about side effects of Librela. Yet CAR$[0, +10]$ is negative after 14 of the 21 events. The largest fall, -15.67 percentage points, followed the European Commission's approval of Portela, which was granted on October 27, 2025 and made public on October 29, an event annotated as favorable. The ZTS mean therefore mixes favorable and adverse news, and the ontology does not record what else happened in the same windows, so the mean cannot be read as an effect of approvals.

6.9 What the review changed in the results

Section 4.7 showed that the review changed which events enter the case study and on which dates. Table 6.6 shows what that does to the results. It reruns the case study, with the same prices and the same code, on two sets of dates that a study without this review could have used. The first puts each final event on the day it happened instead of the day it became public. The second takes the original records as first collected, each on its original date, including the records the review excluded. The pooled row is the mean CAR over the eight companies weighted by event days, a description rather than a pooled test.

Table 6.6: *Mean CAR*$[0, +10]$ *in percentage points with and without the review. Reviewed uses the public dates of the case study, Event date puts each final event on the day it happened, and Original uses the records before the review. R is the number of events analyzed. Values of* p_{shift} *below 0.05 are in bold.*

	Reviewed			Event date			Original		
Ticker	R	CAR	p_{shift}	R	CAR	p_{shift}	R	CAR	p_{shift}
ZTS	21	-2.97	**0.012**	21	-3.28	**0.004**	22	-2.13	0.092
ELAN	18	-2.58	0.230	18	-2.41	0.250	17	-1.62	0.440
PAHC	5	-5.47	0.226	6	-4.58	0.263	5	-4.28	0.282
MRK	23	-1.14	0.281	23	+0.05	0.965	24	+0.46	0.637
IDXX	5	-0.06	0.987	5	-0.06	0.987	5	-0.06	0.987
NEOG	6	-1.42	0.727	6	-1.42	0.727	6	-1.28	0.714
CRL	8	+2.83	0.315	7	+2.02	0.507	7	-0.38	0.892
NOTV	11	-9.70	0.176	11	-9.63	0.198	11	+3.40	0.602
Pooled	97	-2.54	–	97	-2.47	–	97	-0.63	–

Using the date an event happened instead of its public date barely moves the pooled mean, from -2.54 to -2.47 percentage points, but it changes single companies: the MRK mean goes from -1.14 to $+0.05$. The original records change much more. The pooled mean shrinks to -0.63 percentage points, NOTV turns from -9.70 to $+3.40$, CRL turns from $+2.83$ to -0.38, and ZTS, at -2.13, is no longer unusual against its placebo distribution ($p_{\text{shift}} = 0.092$). With identical statistical code, a study on the original records would have reported different results, and nothing in those results would show why. The comparison

does not show that the reviewed results are the true market response. It shows that the results depend on the event records, which is why every record must be open to checking.

One NOTV event shows how a single date changes a result. The original record dated criminal charges against Inotiv's principal supplier of nonhuman primates to November 1, 2022, a month-only date. The review found that Inotiv learned of the charges on November 16 and disclosed them in an SEC filing on November 17. That day the stock closed at $6.82, down from $15.85, a fall of 57.0%, while SPY fell 0.3%. The market model predicted a fall of 0.2%, so the abnormal return was about 56.8 points; in the log returns the study uses, it is -84.17 percentage points. A window that starts on the original date runs from November 1 to November 15 and ends before the fall, with a CAR$[0, +10]$ of -32.54; the reviewed window starts on November 17 and gives -107.59. Without this one event, the reviewed NOTV mean would be $+1.18$ instead of -9.70, and the pooled mean -1.40 instead of -2.54.

6.10 RQ3: repeatable output

Repeatability is judged separately from the financial results. The price snapshot has 27,189 rows and a tracked SHA-256 manifest. Notebook 3 reads no clock and makes no network calls. It records the exact events selected, the identity of the full price data, exclusions, totals, sensitivity subsets, and the comparison without the review. A rerun on the tracked inputs reports every output file as unchanged, byte for byte. As long as the tracked inputs stay the same, the event lists and the tables reproduce exactly.

6.11 RQ4: sensitivity to event definition

Table 6.7 changes one ontology-based filter at a time. The mean is descriptive: it weights each company's CAR by its number of event days. It is not a pooled causal estimate, and it does not model dependence across companies or dates.

In Table 6.7, *Selected* counts the events a definition picks within the analysis period and

Table 6.7: *Sensitivity to ontology-backed subset definitions*

Definition	Selected	Analyzed	Days	Mean CAR
All case-study events	97	97	93	-2.54
Approval-class events	43	43	42	-3.17
Non-approval events	54	54	51	-2.02
Favorable annotation	85	85	82	-1.21
Adverse annotation	10	10	9	-16.24
Neutral annotation	2	2	2	+4.66

Analyzed those with complete estimation and event windows. The two agree in every row because every selected event has complete windows (Section 4.7).

Approval-class events have a mean CAR$[0, +10]$ of -3.17 percentage points, compared with -2.02 for other events. The favorable-annotation subset holds 82 of the 93 event days, the adverse subset nine, and the neutral subset two. The adverse mean is -16.24 percentage points, but the annotations are the researcher's own labels, are very unevenly spread, and are tied to event type.

RQ4 can therefore be answered only as far as the data allow. Changing the event definition clearly changes which events count, how many there are, and the descriptive results. It does not show that any class or annotation causes a particular market response.

6.12 Result summary

The evaluation gives a positive result for the system and an exploratory result for the finance. The complete review matches the final Turtle file, the fixed inputs regenerate the same outputs, and every estimate can be traced back to reviewed events. ZTS shows one unusual placebo result, but no company has enough non-overlapping clusters for standard inference, and no adjusted test was planned in advance. Rerun on the original records, the same code gives different results and no unusual company, so the review shapes the results and not only the records. The evidence supports ontology-backed event modeling as a way to make this kind of study auditable, but not a causal claim about abnormal returns.

Chapter 7

Discussion and conclusions

7.1 Answers to the research questions

RQ1 asked whether every record follows the rules. All 162 final events pass the SHACL checks, match the verification ledger, and pass the custom cross-field checks. Required fields are present, event types match their classes, excluded originals are gone, and public-disclosure dates agree with eligibility. This shows that the data is consistent with the contract, but not that the contract covers every concept in the domain.

RQ2 asked how event dates and public-disclosure dates affect which events can be analyzed. Keeping the two dates apart matters. 49 final events were made public on a different day from the day they happened, and only events with a confirmed public date can enter an analysis, which leaves 2 events as facts only. In the case study, 48 of the 97 analyzed events are analyzed on a different date than the original record gave (Section 6.7). Those dates also change the results: rerun on the original records, the case study gives a pooled mean CAR$[0, +10]$ of -0.63 instead of -2.54 percentage points, and ZTS is no longer unusual (Section 6.9).

RQ3 asked whether the same inputs always give the same outputs. Notebook 3 reads a tracked configuration, the final Turtle records, the verification ledger, the price

file, and its manifest. It checks that the price file's hash matches, records the exact events, builds identifiers and seeds from the content, and produces identical files every time it runs. Notebook 1 rebuilds the Turtle file from the ledger on every run and reports that the result matches the tracked file. This supports computational reproducibility for the saved inputs. It cannot promise that an outside price provider will keep the same historical values.

RQ4 asked how the results change with the event definition. Approval and non-approval definitions change both which events are included and the descriptive mean CAR, and directional-annotation filters produce very uneven samples. The way events are defined clearly matters and should be reported with any result. These small, confounded subsets cannot support claims about the effect of an event category.

7.2 What "ontology backed" means here

In AgriQuant, *ontology backed* describes a real dependency. Whether an event can enter the event study is decided from RDF statements whose classes, properties, and allowed values are defined in an OWL schema and checked by SHACL. The same identifiers and fields then appear in the generated results and in the web application. The ontology is part of the working system, not a diagram drawn beside a separate table.

What the ontology adds is the working chain from meaning, to evidence, to selection, to result. An acquisition cannot quietly turn into a regulatory approval without breaking the class and type rules. An excluded or fact-only record cannot slip into the event study just because it has a date. Every result shows the event date, the disclosure date, and the exact records used, so a reader can check the source and question the decision. AgriQuant shows one complete ontology-backed implementation in which these rules are executable and every result is tied to its reviewed records.

7.3 How reviewed inputs shape the result

Unreviewed inputs can make results look more precise than they are. Treating first-of-month placeholders as exact dates, counting records on the same trading day as separate events, accepting incomplete windows, reading from a price cache that keeps changing, and comparing a result with a single random set of dates each have this effect without any change to the statistical code. None of these can support a research claim.

Holding back every unchecked record has a cost of its own. Of the 97 analyzed events, 67 come from original records whose dates had not been checked before the review, so holding back every unchecked record would have removed most of the sample. Trusting the records already marked as checked would not have been safe either, because the review changed the date of 13 of those 45 (Section 4.7). A record-by-record review avoids both problems. It gives every original record a decision, keeps disclosure dates apart from event dates, and produces a fixed sample of 97 events analyzed with checked prices, complete windows, and clear counts of event days and clusters. The result follows from the reviewed evidence, whichever way it points.

The ZTS pattern should not be generalized to animal-health news as a whole. It is one unadjusted check among several companies and windows, based on seven clusters from a selected corpus. It suggests only that this sample deserves a follow-up study with an independent review and a design fixed in advance.

7.4 Threats to validity

7.4.1 Corpus coverage and selection

The corpus grew from a sample that was not random, and coverage is uneven across companies and event classes. The sources that were available affected what was included. The full review settles the sampled records but cannot fill in what was never sampled. The descrip-

tive CAR values should not be read as averages for the whole sector. The eight case-study companies also differ in how much of their business is animal health, from Zoetis, which sells mainly animal-health products, to Merck, where animal health is about 9% of sales (Section 6.1).

7.4.2 Event timing

A public-disclosure date is better than a month placeholder or a hidden decision date, but it is not always the moment the market first learned the news. Information can leak, and a source may formalize something said earlier. The ontology also does not record time of day or time zone. Moving each date to the first trading day on or after disclosure can place an after-hours announcement one session off.

7.4.3 Other news and causation

The market model removes moves shared with the overall market. It does not remove earnings announcements, economy-wide shocks, lawsuits, product news, or other company announcements near an event. Clustering handles overlap between the included events, but it does not find or remove news that was left out. The study reports what happened around disclosures, not a causal effect.

7.4.4 Market data

The snapshot holds the daily closes reported by Stooq. This research checks the file's identity, date range, and row counts, but it does not check how splits and dividends were handled against a licensed total-return source. Different adjustment methods can change log returns. The results are reproducible for the saved snapshot, not across every data vendor.

7.4.5 Statistical power and multiple comparisons

Each company has between three and twelve non-overlapping clusters. That is too few for standard cluster-robust inference, so intervals and t-tests are not reported. The circular-shift values are exploratory. They do not fix low power, multiple companies, multiple windows, or every form of dependence over time and across companies. A confirmatory study would need a design fixed in advance, an adjustment for multiple comparisons, and many more independent clusters.

7.4.6 Review and annotation

The review leaves every source, reason, and decision open to audit. The researcher carried it out with AI assistance, and no second domain expert repeated it, so there is no measure of agreement between reviewers. The favorable, adverse, and neutral labels record the researcher's judgment, not the actual direction of the stock. Most events are labeled favorable, and the labels are tied to event type. The directional annotation is metadata to inspect, not a tested signal.

7.4.7 Ontology evaluation and persistence

Passing the checks shows that the data follows the rules, and the competency-question queries show that the ontology answers the questions it was designed for. Neither shows that the ontology is complete, that these are the right questions, or that it helps the people who use it. Discussions with the agricultural-finance expert shaped the direction of this research, but it includes no formal expert rating and no controlled comparison with other ways of managing the same data. The namespaces point to files in the project repository. They are not yet publicly resolvable identifiers or DOI-backed publications, and the ontology is not served with content negotiation.

7.5 Lessons for research and engineering

This research points to three practical lessons. First, keep the date an event happened apart from the date it was made public, and make the public date part of the rule for what can be analyzed. Second, report counts separately, because original records, final events, analysis-ready events, analyzed events, distinct event days, and non-overlapping clusters each answer a different question. Third, a result is not reproducible if the events are versioned but the prices or the selection logic can still change.

The design also gives each part one job. OWL describes the concepts, SHACL checks the shape of the stored graph, the rules in notebooks 1 and 2 check the ledger and the cross-field consistency, one selection function decides eligibility, and the event-study code owns the statistical units and output identity. Keeping these boundaries clear means the selection logic is written once, and the web application only shows what the notebooks saved.

7.6 Future work

The first priority is an independent second review, not a more complex model. A second domain expert should repeat a stratified sample or the whole ledger review, settle disagreements, and report agreement on event identity, class, type, event date, disclosure date, and directional annotation. Time of day and time zone should also be recorded separately from the calendar date.

The second priority is better data and study design. A documented total-return price source should replace or check the current closing prices. Rules for overlapping news, overlapping company announcements, dependence across companies, the windows to test, and the adjustment for multiple comparisons should all be set before looking at results. A confirmatory event study should be preregistered only once there are enough independent clusters with confirmed disclosure dates.

Better data would also allow the cross-company question that the agribusiness companies

were collected for: whether animal-health events move agribusiness stocks. That study needs a relation in the ontology stating why one company's news should affect another company's stock, such as a supplier or customer link, which the current model leaves out (Section 5.4). It also needs many more agribusiness events than the three per company that the corpus holds now.

The third priority is evaluating the ontology with the people who would use it. The competency questions are executable queries, so domain reviewers can judge whether they are the right questions and whether the answers help them. A controlled comparison with a constrained CSV, JSON Schema, or relational version could measure curation errors, traceability, time to complete tasks, and how well reviewers understand the data. That would measure what the ontology adds in day-to-day curation.

Interoperability can then improve through alignment with SEM, LODE, FIBO, PROV-O (the World Wide Web Consortium's ontology for describing provenance), and outside identifiers for companies, products, active ingredients, regulators, and source documents.[26, 27, 29, 59] Alignment would add mapping statements in a separate file and leave the event data unchanged. They would state, for example, that every `ag:Event` is a SEM event, that `hasCompany` is a kind of SEM's `sem:hasActor`, and that `eventDate` is a kind of SEM's `sem:hasTimeStamp`. Tools that understand SEM could then read the AgriQuant events directly. The public-disclosure date and the eligibility flag have no counterpart in these vocabularies and would remain AgriQuant's own terms. The HermiT reasoner would then check that the mappings introduce no contradiction.

Machine learning belongs in a separate future study. It would need a defined target and horizon, training, validation, and test periods in time order, protection against leakage, enough data to fit a model, baselines without the ontology, and out-of-sample error. An ontology can organize candidate features, but it does not make predictions valid.

7.7 Conclusion

AgriQuant addresses a question that can be answered and defended: how to model events with an ontology so that financial event studies can be checked and repeated. Its full review turns 169 original records into an RDF corpus of 162 events, with executable rules, separate event and disclosure dates, and one shared path into the analysis. A fixed snapshot of 27,189 prices and identifiers built from content make every reported result repeatable and traceable to the reviewed records.

The review also shows the limits of what the corpus can support. The main case study analyzes 97 events, no company has enough clusters for standard inference, and the ZTS result is an unadjusted, exploratory check. This research therefore claims no forecast, no trading signal, and no causal effect on returns.

Together, the five contributions listed in Section 1.3 form a method that runs. Unsupported records are excluded rather than guessed. Fact-only events cannot slip into the analysis. Event definitions produce clear subsets that can be recovered later. Prices are pinned by their content, and every result carries its source events. The review of the inputs is part of the contribution, because it decides both the sample and what this research can claim.

Bibliography

[1] A. Craig MacKinlay. Event studies in economics and finance. *Journal of Economic Literature*, 35(1):13–39, 1997.

[2] Stephen J. Brown and Jerold B. Warner. Using daily stock returns: The case of event studies. *Journal of Financial Economics*, 14(1):3–31, 1985.

[3] John Y. Campbell, Andrew W. Lo, and A. Craig MacKinlay. *The Econometrics of Financial Markets*. Princeton University Press, Princeton, NJ, 1997.

[4] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, et al. Knowledge graphs. *ACM Computing Surveys*, 54(4):1–37, 2021. doi: 10.1145/3447772.

[5] Nirmal Gelal, Aastha Gautam, Soheil Abadifard, Nico Giordano, Moumita Sen Sarma, Sanaz Saki Norouzi, Claudio Dias da Silva Jr., Jean Ribert Francois, Kathleen M. Jagodnik, Katherine Nelson, Terry Griffin, Xiaomao Lin, Stacy Hutchinson, Stephen M. Welch, Kelsey Andersen Onofre, Romulo Lollato, Pascal Hitzler, and Hande Küçük McGinty. From field data to global food systems intelligence: A semantic graph framework for sustainable wheat production, 2026. arXiv:2502.19507v2.

[6] John H. Davis and Ray A. Goldberg. *A Concept of Agribusiness*. Division of Research, Graduate School of Business Administration, Harvard University, Boston, 1957.

[7] W3C. RDF 1.1 concepts and abstract syntax, 2014. World Wide Web Consortium Recommendation.

[8] W3C. RDF 1.1 Turtle: Terse RDF Triple Language, 2014. World Wide Web Consortium Recommendation.

[9] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[10] W3C. OWL 2 Web Ontology Language document overview, 2012. World Wide Web Consortium Recommendation.

[11] W3C. Shapes Constraint Language (SHACL). W3C Recommendation, 2017. URL `https://www.w3.org/TR/shacl/`.

[12] Hande Küçük McGinty. *KNowledge Acquisition and Representation Methodology (KNARM) and Its Applications*. PhD thesis, University of Miami, 2018. URL `https://scholarship.miami.edu/esploro/outputs/doctoral/KNowledge-Acquisition-and-Representation-Methodology-KNARM/991031447865202976`.

[13] Hande Küçük McGinty, Ubbo Visser, and Stephan Schürer. How to develop a drug target ontology: KNowledge Acquisition and Representation Methodology (KNARM). In *Bioinformatics and Drug Discovery*, volume 1939 of *Methods in Molecular Biology*, pages 49–69. Springer New York, New York, NY, 2019. doi: 10.1007/978-1-4939-9089-4_4.

[14] Tim Bray. The JavaScript Object Notation (JSON) data interchange format. Request for Comments 8259, Internet Engineering Task Force, 2017.

[15] Stooq. Stooq historical market data, 2026. URL `https://stooq.com/q/d/l/`. Provider-reported daily close snapshot archived with the AgriQuant reproducibility artifact.

[16] National Institute of Standards and Technology. Secure Hash Standard (SHS). Federal Information Processing Standards Publication 180-4, U.S. Department of Commerce, 2015.

[17] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain

Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, Carol Willing, and Jupyter Development Team. Jupyter notebooks – a publishing format for reproducible computational workflows. In Fernando Loizides and Birgit Schmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87–90, Amsterdam, 2016. IOS Press. doi: 10.3233/978-1-61499-649-1-87.

[18] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.

[19] W3C. SPARQL 1.1 query language, 2013. World Wide Web Consortium Recommendation.

[20] Michael Grüninger and Mark S. Fox. Methodology for the design and evaluation of ontologies. In *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI-95*, Montreal, Canada, 1995.

[21] Leslie Lamport. *LaTeX: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, 2 edition, 1994.

[22] James W. Kolari and Seppo Pynnönen. Event study testing with cross-sectional correlation of abnormal returns. *The Review of Financial Studies*, 23(11):3996–4025, 2010. doi: 10.1093/rfs/hhq072.

[23] A. Colin Cameron and Douglas L. Miller. A practitioner's guide to cluster-robust inference. *Journal of Human Resources*, 50(2):317–372, 2015. doi: 10.3368/jhr.50.2.317.

[24] Alla A. Golub, Christine A. Wilson, and Allen M. Featherstone. The effects of StarLink corn food safety events on returns and risk of agribusiness firms. Staff Paper 05-08, Department of Agricultural Economics, Purdue University, West Lafayette, IN, 2005.

[25] Grant Gondell. Agriculture hedge funds. Master of Agribusiness the-

sis, Kansas State University, 2019. URL https://krex.k-state.edu/items/03c8c059-c11b-4af7-84e0-f083533a5844. Major professor: Allen M. Featherstone.

[26] Ryan Shaw, Raphaël Troncy, and Lynda Hardman. LODE: Linking Open Descriptions of Events. In *The Semantic Web: Fourth Asian Conference, ASWC 2009*, volume 5926 of *Lecture Notes in Computer Science*, pages 153–167, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. doi: 10.1007/978-3-642-10871-6_11.

[27] Willem Robert van Hage, Véronique Malaisé, Roxane Segers, Laura Hollink, and Guus Schreiber. Design and use of the Simple Event Model (SEM). *Journal of Web Semantics*, 9(2):128–136, 2011. doi: 10.1016/j.websem.2011.03.003.

[28] Simon Gottschalk and Elena Demidova. EventKG: A multilingual event-centric temporal knowledge graph. In *The Semantic Web: 15th International Conference, ESWC 2018*, Lecture Notes in Computer Science, pages 272–287, Cham, 2018. Springer International Publishing. doi: 10.1007/978-3-319-93417-4_18.

[29] EDM Council. Financial Industry Business Ontology (FIBO). https://spec.edmcouncil.org/fibo/, 2026. Enterprise Data Management Council. Specified in RDF(S) and OWL. Accessed July 2, 2026.

[30] Yakov Shafranovich. Common format and MIME type for comma-separated values (CSV) files. Request for Comments 4180, Internet Engineering Task Force, 2005.

[31] Nusrat Rouf, Majid Bashir Malik, Tasleem Arif, Sparsh Sharma, Saurabh Singh, Satyabrata Aich, and Hee-Cheol Kim. Stock market prediction using machine learning techniques: A decade survey on methodologies, recent developments, and future directions. *Electronics*, 10(21):2717, 2021. doi: 10.3390/electronics10212717.

[32] Mahinda Mailagaha Kumbure, Christoph Lohrmann, Pasi Luukka, and Jari Porras. Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications*, 197:116659, 2022. doi: 10.1016/j.eswa.2022.116659.

[33] Jigar Patel, Sahil Shah, Priyank Thakkar, and K. Kotecha. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, 42(1):259–268, 2015. doi: 10.1016/j.eswa.2014.07.040.

[34] Meizir and Brady Rikumahu. Prediction of agriculture and mining stock value listed in Kompas100 index using artificial neural network backpropagation. In *2019 7th International Conference on Information and Communication Technology (ICoICT)*, pages 1–5. IEEE, 2019. doi: 10.1109/ICoICT.2019.8835284.

[35] Zhi-Yuan Chen and Kai Ling Sin. Long short-term memory model based agriculture commodity price prediction application. In *Proceedings of the 2020 2nd International Conference on Information Technology and Computer Communications*, pages 43–49. ACM, 2020. doi: 10.1145/3417473.3417481.

[36] Yan Guo, Dezhao Tang, Wei Tang, Senqi Yang, Qichao Tang, Yang Feng, and Fang Zhang. Agricultural price prediction based on combined forecasting model under spatial-temporal influencing factors. *Sustainability*, 14(17):10483, 2022. doi: 10.3390/su141710483.

[37] Hailei Zhao. Futures price prediction of agricultural products based on machine learning. *Neural Computing and Applications*, 33(3):837–850, 2021. doi: 10.1007/s00521-020-05250-6.

[38] Houjian Li, Xinya Huang, Fangyuan Luo, Deheng Zhou, Andi Cao, and Lili Guo. Revolutionizing agricultural stock volatility forecasting: A comparative study of machine learning and HAR-RV models. *Journal of Applied Economics*, 28(1):2454081, 2025. doi: 10.1080/15140326.2025.2454081.

[39] Dabin Zhang, Xiaoming Li, Liwen Ling, Huanling Hu, and Ruibin Lin. Integrated GCN–BiGRU–TPE agricultural product futures prices prediction based on multi-

graph construction. *Computational Economics*, 66:3927–3955, 2025. doi: 10.1007/s10614-024-10832-w.

[40] Raven Daniel Martin and Haryono Soeparno. Prediction and correlation modeling between global commodity prices and stock prices in the agribusiness sector using the ensemble learning approach. In *Advances in Smart Knowledge Computing*, Studies in Computational Intelligence, pages 145–159. Springer Nature Switzerland, 2026. doi: 10.1007/978-3-032-01133-6_11.

[41] Yuxiao Jiang, Lei Yang, Shipei Du, Yu Sun, and Yuhao Liang. Research on stock prediction of the automobile manufacturing industry and agricultural product processing industry based on the decision tree optimization algorithm. In *Second International Conference on Big Data, Computational Intelligence, and Applications (BDCIA 2024)*, page 108. SPIE, 2025. doi: 10.1117/12.3059629.

[42] Wensheng Wang and Yuxi Liu. A novel framework for agricultural futures price prediction with BERT-based topic identification and sentiment analysis. *Journal of Forecasting*, 44(6):1969–1992, 2025. doi: 10.1002/for.3278.

[43] Jia-Lang Xu and Ying-Lin Hsu. The impact of news sentiment indicators on agricultural product prices. *Computational Economics*, 59(4):1645–1657, 2022. doi: 10.1007/s10614-021-10189-4.

[44] Y. Yang, Z. Wei, Q. Chen, and L. Wu. Using external knowledge for financial event prediction based on graph neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 2161–2164, New York, NY, USA, 2019. ACM. doi: 10.1145/3357384.3358156.

[45] Yinglun Zhang, Antonina Broyaka, Jude Kastens, Allen M. Featherstone, Cogan Shimizu, Pascal Hitzler, and Hande Küçük McGinty. Sustainable grain transportation in Ukraine amidst war utilizing KNARM and KnowWhereGraph. In *Companion*

Proceedings of the ACM Web Conference 2023, pages 742–745, New York, NY, USA, 2023. ACM. doi: 10.1145/3543873.3587618.

[46] Antonina Broyaka, Allen M. Featherstone, Yinglun Zhang, Jude Kastens, and Hande Küçük McGinty. Adapting Ukrainian grain supply chains to the Russia-Ukraine war. *International Food and Agribusiness Management Review*, 29(1):9–37, 2025. doi: 10.22434/IFAMR.1209.

[47] Yinglun Zhang, Aryan Singh Dalal, Caleb Martin, Srikar Reddy Gadusu, and Hande Küçük McGinty. OLIVE: Ontology Learning with Integrated Vector Embeddings. *Applied Ontology*, 20(1):36–53, 2025. doi: 10.1177/15705838251329268.

[48] Yinglun Zhang, Sonia Moavenzadeh, Jarrar Amjad, Onur Apul, Adrita Barua, Fatih Evrendilek, Torsten Hahmann, Ganga Hettiarachchi, Pascal Hitzler, David Kedrowski, Vasu Kilaru, Prayas Lashkari, Katrina Schweikert, Antony Williams, and Hande Küçük McGinty. CompTox ontology: Leveraging knowledge graphs for PFAS monitoring and decision-making. In *Proceedings of the ACM Web Conference 2026*, pages 9352–9360, New York, NY, USA, 2026. ACM. doi: 10.1145/3774904.3792985.

[49] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, et al. The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, 3:160018, 2016. doi: 10.1038/sdata.2016.18.

[50] Birte Glimm, Ian Horrocks, Boris Motik, Giorgos Stoilos, and Zhe Wang. HermiT: An OWL 2 reasoner. *Journal of Automated Reasoning*, 53(3):245–269, 2014. doi: 10.1007/s10817-014-9305-1.

[51] Martin Dürst and Michel Suignard. Internationalized resource identifiers (IRIs). Request for Comments 3987, Internet Engineering Task Force, 2005.

[52] Tim Berners-Lee, Roy T. Fielding, and Larry Masinter. Uniform resource identifier

(URI): Generic syntax. Request for Comments 3986, Internet Engineering Task Force, 2005.

[53] Stanford Center for Biomedical Informatics Research. Protégé Desktop. `https://protege.stanford.edu/`, 2026. Accessed July 12, 2026.

[54] International Organization for Standardization. Information and documentation – digital object identifier system. International Standard ISO 26324:2022, 2022.

[55] Rebecca C. Jackson, James P. Balhoff, Eric Douglass, Nomi L. Harris, Christopher J. Mungall, and James A. Overton. ROBOT: A tool for automating ontology workflows. *BMC Bioinformatics*, 20:407, 2019. doi: 10.1186/s12859-019-3002-3.

[56] Merck & Co., Inc. Form 10-K for the fiscal year ended December 31, 2024. U.S. Securities and Exchange Commission, 2025. URL `https://www.sec.gov/Archives/edgar/data/310158/000162828025007732/mrk-20241231.htm`. Segment sales table: Animal Health $5,877 million of total sales of $64,168 million in 2024.

[57] Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, New York, 1993.

[58] Belinda Phipson and Gordon K. Smyth. Permutation P-values should never be zero: Calculating exact P-values when permutations are randomly drawn. *Statistical Applications in Genetics and Molecular Biology*, 9(1):Article 39, 2010. doi: 10.2202/1544-6115.1585.

[59] W3C. PROV-O: The PROV ontology, 2013. World Wide Web Consortium Recommendation.

Appendix A

Reproduction procedure

This appendix explains how to rebuild the validation results and the analysis from a fresh copy of the repository. Run the commands from the repository root with Python 3.12.

A.1 Install dependencies

```
python3.12 -m venv .venv
.venv/bin/pip install -r requirements.txt
```

This installs JupyterLab and the Python libraries that the notebooks use, at the versions pinned in `requirements.txt`.

A.2 Run the notebooks

```
.venv/bin/jupyter execute --inplace notebooks/01-review-ledger.ipynb
.venv/bin/jupyter execute --inplace notebooks/02-ontology-checks.ipynb
.venv/bin/jupyter execute --inplace notebooks/03-event-study.ipynb
```

The notebooks run in this order. They can also be opened with `.venv/bin/jupyter lab notebooks` and run one step at a time. Every check stops its notebook when it fails, so three notebooks that run to the end have passed every check.

Notebook 1 builds the final Turtle file from the verification ledger and always produces the same output. It fails if any original record lacks a decision, a final identifier clashes or does not trace to an original record, a required field is missing, a controlled value is invalid, a date is not a real calendar date, or eligibility does not match the disclosure date. The expected result is 169 original records, 162 final events, 160 analysis-ready events, 2 fact-only events, and the Turtle file reported as unchanged.

Notebook 2 checks the knowledge graph. It fails if SHACL finds a violation, a class and type disagree, a competency query in Appendix D returns the wrong number of rows, or query 7 does not match the events in the saved results. It then makes deliberate errors in a copy of the graph and confirms that the checks catch them. Its last check needs Java and ROBOT: it merges the schema and event corpus into one ontology, reloads it, and runs the HermiT reasoner, and it fails if the combined graph is inconsistent. Without ROBOT, the notebook says so and skips this one check.

A.3 Verify the market snapshot

The tracked analysis configuration is

`reproducibility/analysis-config.json`

and the archived market files are

`reproducibility/market-prices.csv`

`reproducibility/market-prices-manifest.json`

The CSV contains 27,189 rows. Its expected SHA-256 hash is shown below, and notebook 3 checks the full hash before any analysis.

```
8b8b5b5e9c559ccc6181ebb747dfa7de4167865447f14cb6ecd08f61735ee5b5
```

The manifest also fixes the provider, the meaning of the price field, the symbol list, the date range, and the total and per-symbol row counts. A file with the right shape but different values is rejected because its hash will not match.

A.4 Regenerate analysis output

Notebook 3 reads the final Turtle corpus and the tracked market snapshot, and writes the structured results and generated LaTeX fragments to `results`. It does not contact a market-data provider or read the current date, and it reports every file it writes as unchanged when the tracked inputs are unchanged.

The principal generated files are `event-study-results.json`, which contains the configuration, corpus audit, exact selected events, analysis identities, exclusions, aggregates, subset sensitivity, and the comparison without the review; `analysis-summary.tex`, which contains shared counts and date macros; the table fragments `own-firm-effects.tex`, `own-firm-diagnostics.tex`, `placebo-p-values.tex`, `subset-sensitivity.tex`, and `review-comparison.tex`.

A.5 Check the web application

The web application needs Node.js 24 (the repository pins version 24.1.0 in `.nvmrc`) and npm.

```
npm install
npm test
npm run build
```

The tests load the files that notebooks 1 and 3 saved and check that each view shows them, including the counts from 169 records to 97 analyzed events, the case-study results in the order of Table 6.3, and withheld inference. The production build checks the types in the application's code, which is written in TypeScript, and packages it for the browser. `npm run dev` starts the application on the local machine.

A.6 Build the documents

Both `paper/thesis/thesis.tex` and the companion article `paper/main.tex` build with `latexmk`, a tool that reruns LaTeX until all cross-references are resolved, together with BibTeX, which formats the bibliography. Both tools come with standard LaTeX distributions.

```
cd paper
latexmk -pdf -interaction=nonstopmode -halt-on-error main.tex
```

```
cd thesis
latexmk -pdf -interaction=nonstopmode -halt-on-error thesis.tex
```

The file `thesis.tex` also builds with Tectonic, a self-contained LaTeX engine, by running `tectonic thesis.tex` in `paper/thesis`. It uses the unmodified Kansas State University `class_diss.cls` file.

A.7 Reproducibility boundary

These steps reproduce the archived computation. They cannot show that Stooq's closing prices match a licensed total-return series, that each reviewed disclosure is the moment the market first learned the news, that a second independent reviewer would agree with the ledger, or that the corpus is complete. Those are questions about the data, discussed in Chapter 7, and no repeatable build can answer them.

Appendix B

Event data contract

Table B.1 lists the fields the analysis reads from the RDF graph. "Exactly one" is a rule enforced by the SHACL shapes or, for the identifier and the class, by the checks in notebooks 1 and 2. Optional fields still follow any class-specific rules that apply to them.

Table B.1: Analysis-facing event fields

Field	Cardinality	Form	Meaning and validation
event identifier	Exactly one	Named RDF node	Unique key of the final event. It is opaque; consumers must not parse dates or types from the identifier.
original record identifier	Exactly one	String	Links the final event to the reviewed original record.
ontology class	Exactly one concrete class	RDF type	DrugApproval, PipelineEvent, ScientificDisclosure, RegulatoryAction, CorporateEvent, or ProductLaunch. Concrete classes are disjoint.
company	Exactly one	Object property	Links the event to a controlled company node with ticker and company label.
event date	Exactly one	`xsd:date`	Date of the decision, transaction, incident, launch, or announcement.
public disclosure date	Zero or one	`xsd:date`	Earliest exact contemporaneous disclosure established by review; event-study alignment date.

Continued on next page

Table B.1 continued

Field	Cardinality	Form	Meaning and validation
analysis eligible	Exactly one	Boolean	True exactly when public disclosure date is present.
date kind	Exactly one	String	Meaning of the event date.
title	Exactly one	String	Human-readable event or product description. Nonempty.
event type	Exactly one	Controlled individual	One of the type values defined by the ontology. Each type maps to one expected concrete class.
registry	Zero or one	Controlled individual	Regulator or registry. Required for regulator-backed records under the custom integrity rule.
species	Zero or more	Controlled individual	Target species. Multivalued because one veterinary product can cover multiple species.
directional annotation	Exactly one	Controlled individual	Researcher label `good`, `bad`, or `neutral`, shown in tables as favorable, adverse, or neutral. Metadata only; not source fact, realized return, or validated predictor.
annotation rationale	Exactly one	String	Ex-ante rationale for the directional annotation.
source URL	Exactly one	`xsd:anyURI`	Source supporting the event. Must match an accepted web-address pattern.
source title	Exactly one	String	Human-readable title of reviewed primary evidence.
source publisher	Exactly one	String	Organization responsible for the reviewed evidence.

B.1 Controlled event types

The ontology defines 23 controlled event types. Table B.2 lists each type with the event class it must belong to. The OWL schema maps each type to its event class, and SHACL limits each class to its permitted types.

Table B.2: *Controlled event types and their expected classes*

Event type	Expected class
`regulatory_approval`	DrugApproval
`label_expansion`	DrugApproval
`pipeline_update`	PipelineEvent
`study_publication`	ScientificDisclosure
`scientific_presentation`	ScientificDisclosure
`regulatory_action`	RegulatoryAction
`emergency_use_authorization`	RegulatoryAction
`eua_amendment`	RegulatoryAction
`label_change`	RegulatoryAction
`product_presentation_approval`	RegulatoryAction
`positive_regulatory_opinion`	RegulatoryAction
`method_revocation`	RegulatoryAction
`proposed_approval_withdrawal`	RegulatoryAction
`acquisition`	CorporateEvent
`acquisition_announcement`	CorporateEvent
`divestiture`	CorporateEvent
`corporate_event`	CorporateEvent
`joint_venture`	CorporateEvent
`capacity_expansion`	CorporateEvent
`restructuring`	CorporateEvent
`production_disruption`	CorporateEvent
`product_launch`	ProductLaunch
`product_announcement`	ProductLaunch

B.2 Eligibility state model

Every original record has a final decision of verified or excluded. The final corpus holds only the kept originals and their single-event replacements. A final event is analysis-ready when the review found an exact public-disclosure date. A fact-only event has an exact event date but no exact public announcement from the time. It stays searchable and auditable but cannot enter the event study. Originals that could not be resolved are excluded rather than kept with a made-up date.

B.3 Analysis-count definitions

The software reports several counts because each answers a different question. Table B.3 defines them.

Table B.3: *Analysis-count definitions*

Count	Meaning
Corpus matches	Rows matching the company, ontology-class, and directional-annotation filters, counted before the eligibility gate and before the date interval is applied
Ineligible	Matching fact-only rows without a public-disclosure date
Outside interval	Analysis-ready rows whose public-disclosure date falls outside the analysis interval
Selected rows	Analysis-ready rows whose public-disclosure date falls inside the interval, passed to the event study
Analyzed rows	Selected rows with complete estimation and event windows
Event days	Distinct company trading days after same-day records are merged
Clusters	Connected groups of overlapping event windows used to assess effective independence

These counts should not be used interchangeably. In particular, a large number of corpus rows does not mean a large sample for inference.

Appendix C

Methodological safeguards and rejected practices

This appendix lists weak ways of handling event data that AgriQuant avoids, and the safeguard used for each. Each row names a practice an event-study pipeline could fall into, why it falls short, and what AgriQuant does instead, with its effect on the reported analysis.

Table C.1: Safeguard ledger

Area	Inadequate practice	Implemented safeguard and consequence
Research claim	Ontology-backed work is presented as stock prediction without a predictive protocol, train/test separation, or baseline.	This research evaluates ontology-backed event modeling and reproducible event studies. Machine learning is future work requiring a separate design.
Event dates	Before review, 56 of the 169 original records gave only the month of an event, stored as the first day of that month and treated as exact.	Full review corrected 78 event dates and separates them from public-disclosure dates; 49 final events have different values for the two.

Continued on next page

Table C.1 continued

Area	Inadequate practice	Implemented safeguard and consequence
Ontology consistency	`drugName` was declared only for drug approvals, yet two reviewed events of other classes (an acquisition and a regulatory action) carry a drug name, so OWL reasoning found the merged schema and corpus inconsistent. SHACL could not detect this because it does not apply `rdfs:domain` rules.	The domain of `drugName` covers all events, and the schema declares `xsd:date` explicitly so that HermiT accepts the dates. The merged graph passes the HermiT consistency check, an additional check in the reproduction procedure (Appendix A) that needs Java and ROBOT.
Source coverage	Presence of a URL was treated as adequate provenance.	Every final event requires a source URL, title, and publisher plus a final evidence-backed review decision.
Review completeness	A partial date gate leaves unresolved records pending indefinitely. Before review, 124 of the 169 original records had dates that had not been checked against a source, and the other 45 were trusted because they were marked as checked.	All 169 originals were decided, giving 162 final events (Table 4.1). Of the 97 analyzed events, 67 come from originals whose dates had not been checked, and the review changed the date of 13 of the 45 already-checked records.
Atomicity and duplicates	Composite disclosures and duplicate approvals could remain as event rows.	A record describing several events is excluded and replaced by separate single-event records; a duplicate is excluded with no replacement.
Analysis interval	A rolling five-year window depended on the date when the code ran.	The analysis covers events publicly disclosed from 2021-01-01 through 2025-12-31, and the price interval is fixed to 2020-06-01 through 2026-02-09.
Market input	Reported output could depend on a mutable price cache or a live provider response.	Notebook 3 reads a tracked 27,189-row CSV file and fails unless its manifest and SHA-256 digest match.
Window completeness	Events near a price boundary could be analyzed with less than the declared history or event window.	Every event requires exactly 120 estimation observations and the complete $[-10, +20]$ event window.

Continued on next page

Table C.1 continued

Area	Inadequate practice	Implemented safeguard and consequence
Nontrading dates	Calendar dates were not consistently tied to a documented common-market convention.	Events align to the first common stock/SPY trading date on or after the verified public-disclosure date.
Same-day records	Multiple rows aligned to the same firm/trading date could be counted as independent events.	They are one statistical unit while all source identifiers remain attached.
Overlapping windows	Event windows could overlap without the reported effective sample reflecting the dependence.	Connected overlaps form clusters; tables report both event days and clusters.
Few-cluster inference	Conventional intervals and p-values could be shown for very small effective samples.	Cluster inference is withheld below 20 clusters. Every firm in the primary case study is below that threshold.
Placebo control	One random-date comparison could be interpreted as evidence that the result was unusual.	Deterministic circular shifts generate a reference distribution while preserving within-firm event spacing. ZTS is flagged as an exploratory screen, with few-cluster and multiplicity caveats.
Significance language	Screening values were described with statistical-significance language unsupported by adequate controls.	This research separates descriptive CAR, placebo diagnostics, and unavailable conventional inference.
Runtime deduplication	A hidden heuristic could remove approval-like rows during query execution.	Duplicate resolution is a curation and validation responsibility. Statistical same-day collapse remains explicit and is reported separately.
Directional-annotation overrides	The interface could change the ontology's directional annotation through a separate override path.	The annotation in the Turtle file is authoritative, and a missing annotation or rationale fails validation. Nothing in the notebooks or the web application overrides it, and this research labels the field as researcher-authored metadata.

Continued on next page

Table C.1 continued

Area	Inadequate practice	Implemented safeguard and consequence
Spillover analysis	Events could be applied to other securities without a modeled exposure relation.	Events are matched only to their own company's stock, and no spillover table or claim is reported.
Displayed event set	A display could rebuild the event list separately from the calculation.	The web application shows the saved results, which include the exact `eventsUsed` array, and computes nothing itself.
Missing statistics	Null interval or probability values could be formatted as numeric zero in the interface.	Missing inference is displayed as unavailable with the reason, including the cluster threshold.
Reported tables	Numeric values could be manually copied into prose or tables.	The JSON results, the case-study table rows, and the counts in `analysis-summary.tex` are generated from the same fixed analysis, and every rerun checks that they regenerate byte for byte. The remaining counts in the text are written by hand and can be checked against the notebook outputs and the ledger (Section C.1).

C.1 Interpretive consequence

Together, these safeguards make the results follow the reviewed evidence, whichever way it points. The record-by-record review gives a fixed sample of 97 analyzed events and one exploratory ZTS pattern, and standard inference stays unavailable below the cluster cut-off. The case-study tables (Tables 6.3 to 6.7) and the counts that the text takes from the generated file `results/analysis-summary.tex` are generated outputs of the fixed analysis. Other numbers, such as those in Tables 4.2 and 6.2, are written into the text and can be checked against the outputs of notebooks 1 and 2. The review-effect counts in Section 4.7, such as the 45 records marked as checked before the review, are counted from the verification ledger.

Appendix D

Competency-question queries

These are the SPARQL queries for the seven competency questions in Chapter 3, printed from the same files that notebook 2 runs. Each file starts with its question and the number of rows it should return. Chapter 6 reports the answers.

```
# CQ1: Which company does the event belong to?
# expect: 162 rows
PREFIX ag: <https://github.com/progami/ag-econ/blob/main/ontology/ag-econ.ttl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?event ?ticker ?companyName
WHERE {
  ?event a ?class .
  ?class rdfs:subClassOf ag:Event .
  ?event ag:hasCompany ?company .
  ?company ag:ticker ?ticker ;
           ag:companyName ?companyName .
}
ORDER BY ?event
```

```
# CQ2: Which event class and controlled event type describe it
# (for example, class CorporateEvent and type acquisition
# when one company buys another)?
# Joining on expectedEventClass keeps only events whose type matches
# their class.
# expect: 162 rows
```

```
PREFIX ag: <https://github.com/progami/ag-econ/blob/main/ontology/ag-econ.ttl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?event ?class ?eventType
WHERE {
  ?event a ?class .
  ?class rdfs:subClassOf ag:Event .
  ?event ag:hasEventType ?type .
  ?type rdfs:label ?eventType ;
        ag:expectedEventClass ?class .
}
ORDER BY ?event



# CQ3: What date is given for the event itself, and what kind of date is it?
# expect: 162 rows
PREFIX ag: <https://github.com/progami/ag-econ/blob/main/ontology/ag-econ.ttl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?event ?eventDate ?dateKind
WHERE {
  ?event a ?class .
  ?class rdfs:subClassOf ag:Event .
  ?event ag:eventDate ?eventDate ;
         ag:dateKind ?dateKind .
}
ORDER BY ?event



# CQ4: Which primary source supports the event, and when was it first
#      made public?
# Fact-only events have no public-disclosure date, so that column is optional.
# expect: 162 rows
PREFIX ag: <https://github.com/progami/ag-econ/blob/main/ontology/ag-econ.ttl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?event ?sourceUrl ?sourceTitle ?sourcePublisher ?publicDisclosureDate
WHERE {
  ?event a ?class .
  ?class rdfs:subClassOf ag:Event .
  ?event ag:sourceUrl ?sourceUrl ;
         ag:sourceTitle ?sourceTitle ;
         ag:sourcePublisher ?sourcePublisher .
```

```
  OPTIONAL { ?event ag:publicDisclosureDate ?publicDisclosureDate }
}
ORDER BY ?event



# CQ5: Is there a regulator or a target species?
# Both are optional; an empty value answers "no".
# expect: 162 rows
PREFIX ag: <https://github.com/progami/ag-econ/blob/main/ontology/ag-econ.ttl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?event ?registry
       (GROUP_CONCAT(?speciesLabel; separator=", ") AS ?species)
WHERE {
  ?event a ?class .
  ?class rdfs:subClassOf ag:Event .
  OPTIONAL { ?event ag:hasRegistry ?r . ?r rdfs:label ?registry }
  OPTIONAL { ?event ag:targetSpecies ?s . ?s rdfs:label ?speciesLabel }
}
GROUP BY ?event ?registry
ORDER BY ?event



# CQ6: Which directional annotation was assigned, and why?
# expect: 162 rows
PREFIX ag: <https://github.com/progami/ag-econ/blob/main/ontology/ag-econ.ttl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?event ?annotation ?rationale
WHERE {
  ?event a ?class .
  ?class rdfs:subClassOf ag:Event .
  ?event ag:hasSentiment ?sentiment ;
         ag:annotationRationale ?rationale .
  ?sentiment rdfs:label ?annotation .
}
ORDER BY ?event



# CQ7: Which exact events went into a reported number?
# Rebuilds the case-study sample: the eight primary tickers, events with a
# public-disclosure date in the 2021 to 2025 analysis period.
# Notebook 2 also checks these events against eventsUsed in the saved results.
# expect: 97 rows
```

```
PREFIX ag: <https://github.com/progami/ag-econ/blob/main/ontology/ag-econ.ttl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?event ?ticker ?publicDisclosureDate
WHERE {
  VALUES ?ticker { "ZTS" "ELAN" "PAHC" "MRK" "IDXX" "NEOG" "CRL" "NOTV" }
  ?event a ?class .
  ?class rdfs:subClassOf ag:Event .
  ?event ag:hasCompany ?company .
  ?company ag:ticker ?ticker .
  ?event ag:analysisEligible true ;
         ag:publicDisclosureDate ?publicDisclosureDate .
  FILTER (?publicDisclosureDate >= "2021-01-01"^^xsd:date
          && ?publicDisclosureDate <= "2025-12-31"^^xsd:date)
}
ORDER BY ?ticker ?publicDisclosureDate ?event
```