A Relationship-Rich Financial Dataset for Text-to-RDF-Triple Extraction: SEC Fund Disclosures as a Knowledge-Graph Source

Dr. Florian Herzog
Thesis supervisor — companion technical note to the thesis *Magical RDF Triples and how to synthetize them*

June 3, 2026

Abstract

This note specifies a finance-domain dataset for training and evaluating models that extract Resource Description Framework (RDF) triples from plain text, the task at the centre of the accompanying thesis. The dataset is constructed from mandatory U.S. Securities and Exchange Commission (SEC) fund disclosures. Unlike the Wikidata-derived corpora commonly used for this task — where the source text is written *from* the triples and is therefore roughly the same size as its target — here a long natural-language prospectus (on the order of 10^5–10^7 characters) maps to a compact graph of a few hundred triples, yielding a realistic text-to-output size ratio of roughly 20 : 1. Crucially, the target is a genuine graph of *entity-to-entity relationships* (a fund *advised by* a management company, *distributed by* an underwriter, *holding* a security *issued by* an issuer), not a flat list of literal attributes. Two distinct ground-truth regimes are available: a *model-free gold* baseline derived from parallel structured SEC filings (N-CEN, N-PORT, Series/Class reference data), and a *strong-model silver* baseline for the relations expressed only in prose. We describe the source filings, the target ontology and graph structure, the holdings sub-graph, the serialization into the thesis's grammar-terminal token format, and how the resulting samples are used to train and benchmark the four models under study.

Contents

1 Motivation: the size-ratio and relationship gap

The thesis trains a general-purpose language model to extract serialized RDF triples from plain text conditioned on an ontology. The quality of such a model is bounded by the quality of its training data. The benchmarks surveyed in the thesis (WebNLG, T-REx, REBEL, Wiki-NRE) share two properties that make them weak proxies for the real extraction problem.

Symmetric size. In WebNLG, human annotators were instructed to write text *from* a given set of triples. Consequently each sentence encodes almost exactly the triples it was generated from: the input text and the target JSON are of comparable length. The task degenerates towards transliteration and never exercises the central difficulty of practical information extraction — locating a small number of facts inside a large, noisy document.

Attribute-only targets. Many relation-extraction corpora reduce to mapping a sentence to a single predicate label, or to a star of literal-valued attributes around one entity. They contain few *entity-to-entity* edges, and therefore exercise little of the graph structure that motivates RDF in the first place.

A suitable dataset must instead satisfy both of the following, simultaneously:

(i) the input text is substantially larger than the target serialization, so the model must perform genuine reading comprehension over a long document;

(ii) the target is a multi-entity-type graph of relationships, so the inferred ontology contains edges of the form `TypeA --predicate--> TypeB`, not only `TypeA --predicate--> literal`.

SEC fund disclosures satisfy both, and additionally provide a rare third property: a *free, non-model ground truth*, because the same facts that appear in the prose are independently filed by the same registrants in structured form.

2 Source filings

The dataset draws on four public SEC data sources, summarised in Table 1. Their division of labour is the key design idea: the *prose* filings provide the model input, while the *structured* filings provide the ground-truth graph.

Table 1: SEC source filings and their role in the dataset.

Source	Form	Content	Role
Prospectus	N-1A (485BPOS, 497)	Investment objective, strategy, management, fees (prose)	**input text**
N-CEN	N-CEN	Service providers, classification	**gold edges**
N-PORT	NPORT-P	Portfolio holdings (quarterly)	**gold edges**
Series/Class CSV	—	Trust/Series/Class identity	**gold skeleton**
Annual report (MDFP)	N-CSR	Top-holdings commentary (prose)	**input text (holdings)**

Prospectus (N-1A). The statutory prospectus is a long legal document describing a fund family. It names, in prose, the fund's investment adviser, sub-adviser, distributor, transfer agent, portfolio managers and benchmark index, together with its objective, strategy and fee structure. A single filing covers all funds (series) of a trust and ranges from roughly 4×10^5 to 1×10^7 characters of extracted text.

N-CEN. The annual census filing reports, in structured tabular form, each fund's service providers — adviser, sub-adviser, custodian, transfer agent, administrator — and the trust's principal underwriter, each with a Legal Entity Identifier (LEI) where available. These rows are the gold standard for the service-provider edges of the graph.

N-PORT. The monthly portfolio filing reports, per fund, every security held, with issuer name, identifiers (CUSIP, ISIN, LEI), asset category, investment country and market value. These rows are the gold standard for the holdings sub-graph (Section 4).

Series/Class reference data. The SEC's Series/Class listing provides the trust → series → share-class identity backbone, gold for the structural `seriesOf` and `hasShareClass` edges.

A central property is *redundancy across modality*: a fact such as "the fund is advised by Geode Capital Management, LLC" appears both as a sentence in the prospectus (the input) and as a structured row in N-CEN (the label). This is what makes a model-free ground truth possible.

3 Target ontology and graph structure

The target of each sample is a directed, labelled multigraph $G = (E, R)$ in the sense of the thesis, where nodes are typed entities and edges are RDF predicates. The entity types and relations are listed in Table 2.

Table 2: Target ontology: entity types and entity-to-entity relations.

Subject type	Predicate	Object type	
Fund	`seriesOf`	Trust	
Fund	`advisedBy`	InvestmentAdviser	
Fund	`subAdvisedBy`	SubAdviser	
Fund	`transferAgent`	TransferAgent	
Fund	`custodian`	Custodian	
Fund	`administrator`	Administrator	
Trust	`underwrittenBy`	Distributor	
Fund	`holds`	Security	(holdings sub-graph)
Security	`issuedBy`	Issuer	
Security	`domiciledIn`	Country	
Fund	`tracksIndex`	Index	

Every relation in Table 2 has an entity as its object, not a literal. The dataset may optionally be enriched with literal-valued attribute triples (management fee, net expense ratio, returns, portfolio turnover) drawn from the XBRL Risk/Return filings; these are deliberately *secondary*, because the purpose of the dataset is to exercise relational structure.

Following the thesis's ontology-inference procedure (SPARQL meta-schema extraction), the per-sample ontology presented to the model is the set of distinct `(subject type, predicate, object type)` patterns realised in that sample, e.g.

```
{
  "Fund": {
    "seriesOf":       ["Trust"],
    "advisedBy":      ["InvestmentAdviser"],
    "subAdvisedBy":   ["SubAdviser"],
    "transferAgent":  ["TransferAgent"],
    "custodian":      ["Custodian"],
    "administrator":  ["Administrator"]
  },
  "Trust": { "underwrittenBy": ["Distributor"] }
}
```

Listing 1: Inferred ontology for one fund trust (model input, abbreviated).

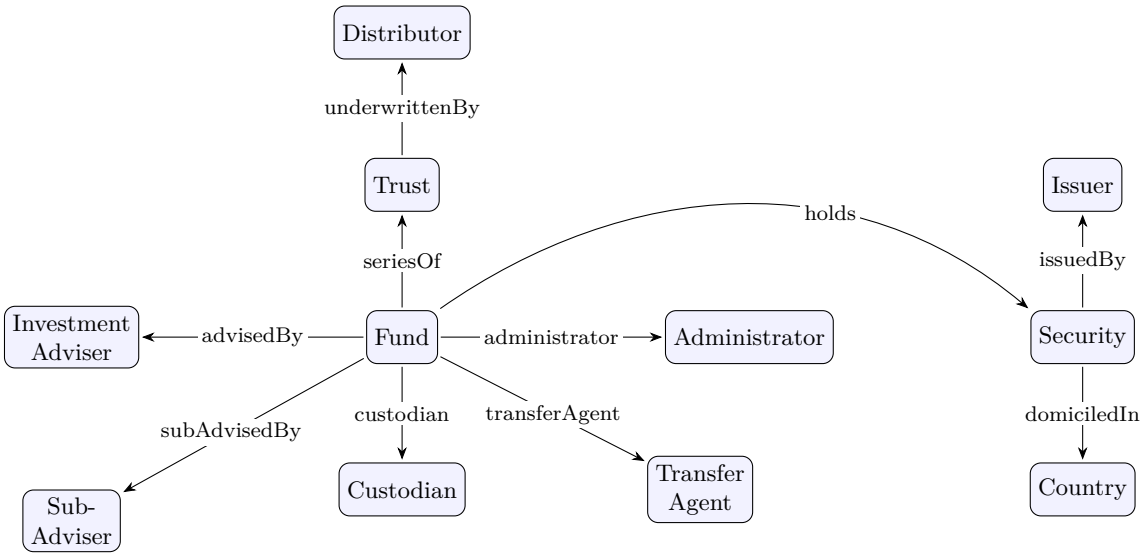

Figure 1: Schematic of the target knowledge graph. Left and centre: the service-provider/structure graph grounded in the prospectus prose. Right column (Issuer–Security–Country): the holdings sub-graph grounded in annual-report commentary with N-PORT gold.

4 The holdings sub-graph

Portfolio holdings express the richest relationships in the data — a fund *holds* many securities, each *issued by* an issuer *domiciled in* a country — and are the natural place to grow the graph beyond service providers. They require care, however, because holdings are *not* disclosed in the prospectus: the prospectus describes a fund's *strategy* ("invests in large-capitalisation equities"), never its specific positions.

The text-bearing source for holdings is the **annual or semi-annual report** (Form N-CSR). It contains two parts:

- the *Schedule of Investments*, a complete table of every holding — structured, not prose, and therefore not an extraction target; and
- the *Management Discussion of Fund Performance* (MDFP), a narrative in which the portfolio manager names the fund's *top* positions and explains their contribution ("our largest holdings were Apple, Microsoft and . . . ").

The MDFP is genuine prose and yields real `holds` edges for the named positions. The corresponding **N-PORT** filing provides the structured gold: the full holdings table, against which the MDFP-named subset can be verified and from which `issuedBy` and `domiciledIn` are taken.

This produces a second, independent text-to-graph task in the same financial domain: *MDFP commentary → holdings sub-graph*, with N-PORT as gold. Because it pairs a *different* document

type with a *different* relation set, including it strengthens the cross-domain generalization claim of the thesis (Section 3.2.3): a single model is shown to extract two structurally different graphs in the same domain. Fund fact sheets and portfolio-manager commentaries published by fund companies are an additional, off-EDGAR prose source for the same edges, at the cost of having no standardized machine-readable gold.

A practical caveat applies to holdings as it does to service providers (Section 7): only the positions *named in prose* are recoverable from the input. The benchmark therefore scopes the `holds` target to the MDFP-named subset rather than the full N-PORT schedule, to avoid penalising a model for failing to extract facts that are absent from its input.

5 Serialization and the marker token format

Targets are serialized in the grammar-terminal token format introduced in the thesis (Section 5.2), in which four special tokens delimit triple components and shared subjects/predicates are factored out, mirroring Turtle's predicate-object lists:

```
<triple_start> Small Cap Special Values Fund
  <predicate_marker> seriesOf
    <object_marker> VALIC Co I
  <predicate_marker> advisedBy
    <object_marker> The Variable Annuity Life Insurance Company
  <predicate_marker> subAdvisedBy
    <object_marker> SunAmerica Asset Management, LLC
  <predicate_marker> administrator
    <object_marker> SunAmerica Asset Management, LLC
  <predicate_marker> custodian
    <object_marker> State Street Bank and Trust Company
<triple_end>
```

Listing 2: Target serialization for one segmented fund (primary-custodian scope).

To support the four-model comparison from a single dataset, each sample carries *two* serializations of the identical triples: `target_serialized`, the marker form above (for Models 2/4, whose vocabulary is extended with the four grammar-terminal tokens), and `target_serialized_plain`, a Turtle-like form using ordinary ';' and ',' delimiters and no special tokens (for Models 1/3). Because the two differ only in whether the delimiters are dedicated tokens, the comparison isolates exactly the effect under study in research question 1.

Each sample is thus a JSON record with: the input prose (`input_text`), the inferred ontology (`ontology`), the target triples as a structured list (`target_triples`) and in both serializations, the trust/series identifiers, and size statistics.

6 Per-fund segmentation

A single prospectus filing covers *all* funds of a trust and may exceed 10^7 characters, beyond any practical context window. Treating one filing as one sample is also semantically wrong: the target would mix the subgraphs of dozens of unrelated funds. The dataset therefore segments each filing into *per-fund* samples, so that one fund's prospectus section maps to that one fund's subgraph.

Fetching all books of a trust. Large fund families split their funds across *several* prospectus books, so the single most recent filing covers only a fraction of a trust's funds. The fetcher therefore retrieves the most recent *full* prospectuses (forms 485BPOS/485APOS) for each trust — preferring them over the much shorter 497/497K supplements, which are used only as a fallback — and concatenates their text. On the proof-of-concept slice this raised the fetched

text from one book per trust to a mean of seven, e.g. from 5×10^5 to 2.2×10^7 characters for a large ETF trust, so that far more fund sections are present.

Section anchors. Statutory prospectuses open each fund's block with the fund name immediately followed by a summary heading. Filers use several styles, so the segmenter accepts any of: "Fund Summary", "Investment Objective", "Principal Investment Strategies", the ETF objective sentence "The Fund seeks…", or a class/ticker header ("Class/Ticker:…").

Boundary selection and a collapse guard. The segmenter collects *all* anchored heading positions across the concatenated text, sorts them, and cuts each segment from one heading to the next. Because a fund name can also occur in tables of contents and cross-references, naïvely taking the first occurrence collapses segments to a few characters; the segmenter therefore discards any candidate whose resulting segment is shorter than a minimum (1,500 characters) and, for each fund, keeps the longest surviving segment. Each segment is paired with that fund's edges plus the fund-anchored `seriesOf` edge and the trust-level `underwrittenBy` edge.

Name-variant matching. The fund name filed in N-CEN and the heading printed in the prospectus frequently differ in the legal-form suffix — a fund filed as "… Fund" may be headed "… ETF" or "… Portfolio". The segmenter matches on a set of normalized variants (suffix swapped or dropped) rather than the exact N-CEN string.

Coverage and fallback. Where a fund's section cannot be located it is skipped and counted (never silently dropped); where no section in a trust can be located, the builder emits a single whole-trust fallback sample. On the proof-of-concept slice, fetching all books and applying the robust segmenter turns 14 trusts into 141 samples (135 cleanly segmented per-fund plus 6 whole-trust fallbacks), with a per-fund median input of $\sim 3.7 \times 10^4$ characters against a $\sim 6.5 \times 10^2$-character target — a per-fund text-to-JSON ratio with a median near $55:1$. The residual misses are dominated not by segmentation but by an *entity-resolution* gap: some trusts file their prospectuses under a different CIK (or fund brand) than their N-CEN report, so the N-CEN fund names do not appear in the fetched text at all. Closing that gap requires joining on the SEC Series identifier across CIKs rather than fetching more filings of the same CIK, and is left to the full-scale build.

7 Ground truth and baselines

The dataset offers two independent ground-truth regimes.

Model-free gold. For `advisedBy`, `subAdvisedBy`, `transferAgent`, `custodian`, `administrator` and `underwrittenBy`, the labels come directly from N-CEN; for `seriesOf` and `hasShareClass`, from the Series/Class reference data; for `holds`, `issuedBy` and `domiciledIn`, from N-PORT. No model is involved in producing these labels, which makes them an unusually trustworthy reference for a generative-extraction benchmark.

The custodian relation and edge scoping. The custodian relation illustrates a subtlety that any honest benchmark on this data must address. N-CEN reports, for a global fund, not only its *primary* custodian but the entire chain of *foreign sub-custodians* — one bank per market it invests in. These sub-custodians have two damaging properties. First, they are *unextractable*: they appear only in the N-CEN table and essentially never in the prospectus prose (a naive string-match recovers 7% of them), so keeping them as targets asks the model to extract facts absent from its input. Second, they *dominate*: with `IS_SUB_CUSTODIAN=Y` accounting for

88% of custodian rows, they constitute roughly two thirds of *all* edges in the unfiltered graph, inflating both the target size and the training loss with noise. The dataset therefore scopes the custodian relation to the *primary* custodian (`IS_SUB_CUSTODIAN≠Y`, a median of one per fund), which is genuinely prose-grounded — it is named in the prospectus or its Statement of Additional Information (e.g. "State Street Bank and Trust Company serves as custodian"). This single change reduces the corpus from 36,880 to 15,739 edges, all of prose-grounded relation types, and is the configurable default (`-custodian-scope primary`). The full sub-custodian chain remains available in N-CEN as a structured-only relation outside the text-to-triples task. This is a dataset-quality decision of the same kind the thesis notes for T-REx and REBEL, whose non-exhaustive references unfairly penalise correct extractions.

No-model lower bound. A trivial string-matching baseline — emit a gold edge iff the object's name occurs in the prose — establishes a *floor* and measures *how prose-grounded each relation is*. Table 3 reports this on the proof-of-concept slice after primary-custodian scoping, multi-book fetching and per-fund segmentation. Because the baseline requires an *exact substring* match within the fund's *own* section, its recall is a strict lower bound: a fund's adviser, for instance, must be named in that fund's segment under a literal spelling. On the full quarter the adviser is recovered with recall 0.93 and the micro-averaged F_1 reaches 0.79; the recovered custodian recall is 0.63 (up from 0.07 unscoped and 0.37 after scoping alone). The residual gap from 1.0 is attributable to surface-form variation ("State Street Bank and Trust Company" vs. "State Street") that a trained model handles but exact matching does not.

Table 3: No-model string-match baseline on the full 2025 Q3 build, after primary-custodian scoping, multi-book fetching and per-fund segmentation (852 samples). Precision is 1.00 by construction; recall is a strict exact-match lower bound.

Relation	Recall	Gold edges
`advisedBy`	0.93	1,673
`seriesOf`	0.84	1,555
`subAdvisedBy`	0.84	946
`administrator`	0.80	2,066
`transferAgent`	0.72	1,721
`custodian`	0.63	1,761
`underwrittenBy`	0.62	863
micro-average	0.65	6,479

Strong-model silver. For relations that are expressed only in prose and lack a clean structured source — portfolio managers (`managedBy`), the named benchmark index (`tracksIndex`), and MDFP-named holdings — a strong reference model (e.g. GPT-4 or Claude Opus) produces silver labels. Because the structured relations have model-free gold, the silver-labelling model can itself be *measured* on the overlapping gold edges, so its reliability is quantified rather than assumed.

8 Corpus statistics

Table 4 summarises one full quarter (2025 Q3) of the dataset. The N-CEN gold graph (after primary-custodian scoping) holds 15,739 entity-to-entity edges across 435 trusts and 2,421 funds. Fetching all full prospectus books for every trust (2,326 filings across 393 trusts; 42 closed-end or interval funds file no standard prospectus) and applying the robust per-fund segmenter yields 852 samples (659 cleanly segmented per-fund plus 193 whole-trust fallbacks). The segmented

samples have a per-fund median ratio near 117 : 1 (input prose to target serialization), and across all samples the median exceeds 400 : 1 — the inverse of the symmetric-size benchmarks.

Table 4: Corpus statistics for the full 2025 Q3 build. Left: N-CEN gold graph (primary-custodian scope). Right: text-to-triple samples (all prospectus books per trust, per-fund segmentation).

Gold graph (2025 Q3)		Samples (2025 Q3)	
Trust graphs	435	Samples (total)	852
Funds (series)	2,421	segmented per-fund	659
Entity-entity edges	15,739	whole-trust fallback	193
custodian (primary)	3,045	Trusts fetched	393
advisedBy	2,588	Prospectus filings	2,326
Distributors	458	Ratio (median, per-fund)	117 : 1

Train/validation/test split. Partitioned at the trust level by a deterministic hash of the CIK: 655 train, 122 validation, 75 test samples (from 268, 37 and 36 trusts respectively), with *no* trust appearing in more than one split. Multiple quarters and the dropping of ontology subsets (per the thesis's augmentation strategy) expand the corpus further.

9 Use in the thesis experiments

Each sample is a triple (x, σ, y) where x is the prospectus prose, σ is the inferred ontology, and y is the marker-serialized triple graph. The model is trained to compute $y = f_\theta(x, \sigma)$. This dataset feeds the four-model comparison of the thesis directly:

- **Model 1/3** (decoder-only / encoder-decoder, no extra tokens): trained on the plain serialization `target_serialized_plain`.
- **Model 2/4** (with grammar-terminal tokens): trained on the marker serialization `target_serialized`, with the four markers `<triple_start>`, `<predicate_marker>`, `<object_marker>`, `<triple_end>` added to the vocabulary as single tokens, testing research question 1 (do dedicated terminal tokens reduce loss and raise F_1).

Splits. The dataset is partitioned into train/validation/test at the *trust* level (80/10/10), assigned by a deterministic hash of the trust CIK. Splitting by trust rather than by fund prevents leakage: funds of one trust share advisers, distributors and custodians, so a fund-level split would let the model memorise trust-specific entities seen in training and inflate test scores. The builder verifies that no trust appears in more than one split.

Because the dataset's input is far longer than its output and its target is a relational graph, it stresses precisely the capabilities the thesis cares about: long-context reading comprehension and faithful generation of entity-to-entity structure. Evaluation uses triple-level precision, recall and F_1 against the model-free gold, matched on `(subject type, predicate, normalized object label)` so that IRI-slug differences do not create spurious errors. The same metric scores the strong-model silver baseline, giving a like-for-like comparison between the finetuned models and a state-of-the-art prompted extractor.

10 Reproducibility

The dataset is built by two scripts accompanying this note. `build_rdf_dataset.py` has three stages: `gold` parses the local N-CEN flat files into per-trust gold graphs (with `-custodian-scope` to choose primary-only, all, or no custodian edges); `fetch` downloads all recent full prospectus

books per trust from EDGAR (rate-limited, `gzip`-aware, `-max-filings` per trust) and concatenates them; `samples` segments the prose per fund and joins it with the gold into the (x, σ, y) records described above. `score_baseline.py` computes the no-model string-match baseline and scores any strong-model predictions against the gold. All inputs are public SEC filings; no licensing restriction applies to the derived dataset.