Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

Layered Semantic Refinement: A Framework and Evaluation Protocol for Task-Specific Embeddings

Phd. Pedro Emílio Amador Salomão
Independent researcher

peas8810@gmail.com
https://orcid.org/0000-0001-9451-3111

Abstract

Dense embeddings are foundational to contemporary natural language processing, information retrieval, recommendation, and retrieval-augmented generation. Nevertheless, a single general-purpose vector typically superimposes multiple relations—topic, entailment, sentiment, part–whole structure, evidential role, temporality, and domain-specific constraints—within one geometry and one similarity function. This article presents layered semantic refinement (LSR) as a framework and evaluation protocol rather than as a single universally validated algorithm. LSR retains a general encoder as a transferable semantic substrate while explicit modules reorganize, augment, or index its representations for a defined operational objective. The framework covers learned projections, nonlinear adapters, graph message passing, hierarchical aggregation, deterministic rule channels, clustering, multi-view representations, and prompt-conditioned embeddings. We distinguish tokens, token embeddings, contextual token states, pooled text embeddings, prompt tokens, prompt embeddings, and instruction-conditioned embeddings. We further explain when short learned prompt tokens can reduce repeated textual instructions or runtime context, while noting that virtual prompt slots still consume transformer positions and computation. We synthesize evidence from BERT and Sentence-BERT, contrastive representation learning, prompt and prefix tuning, INSTRUCTOR, multi-view document retrieval, late-interaction retrieval, hierarchical multitask learning, Matryoshka embeddings, TabEmb, TAROT, TopicForest, and hierarchical news clustering. We provide a formal model, a taxonomy of refinement mechanisms, one canonical LSR instantiation, a token-aware consumption protocol, and a controlled multi-relation experimental blueprint. Because no benchmark was executed for the present article, all numerical examples are explicitly treated as design scenarios rather than empirical findings. We argue that an

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

embedding should be considered organized only when the intended relations are operationally specified, measured in the original space, shown to survive robustness tests, and attributable to the proposed structural bias through ablation.

Keywords: semantic embeddings; token representations; prompt embeddings; prompt tokens; soft prompt tuning; instruction-conditioned embeddings; multi-view retrieval; late interaction; representation learning; hierarchical learning; graph neural networks; semantic search; retrieval-augmented generation; token efficiency; model governance.

1. Introduction

The transition from discrete lexical features to continuous distributed representations changed the computational treatment of language. In a dense embedding space, linguistic units are mapped to vectors whose relative positions can encode regularities that would be expensive to enumerate symbolically. Contextual encoders subsequently made these vectors conditional on surrounding text, allowing the same surface form to acquire different representations across contexts (Devlin et al., 2019). Sentence-level encoders then optimized independently computed vectors for efficient similarity search and clustering (Reimers & Gurevych, 2019). Today, embeddings function as an infrastructural layer for document retrieval, semantic deduplication, recommendation, classification, topic discovery, and retrieval-augmented generation (RAG).

Operational success, however, can encourage an imprecise assumption: that a semantically rich vector is already organized for every downstream use. A general encoder is trained under broad objectives over heterogeneous corpora. It may place related texts near one another, but it is not obliged to separate the exact subclasses required by a clinical evidence system, preserve a document's section hierarchy, model relations among table columns, or distinguish topical similarity from argumentative agreement. These relations are often superimposed in a common representation and reduced at inference time to a single metric such as cosine similarity.

We use the term functionally flat space to describe this operational limitation. The expression does not assert that the learned manifold is geometrically Euclidean or intrinsically unstructured. Rather, it indicates that multiple relations are delivered through one undifferentiated interface: one vector, one index, and often one similarity function. A pair of sentences can be topically similar yet sentimentally opposed; two biomedical

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

abstracts can discuss the same disease while differing in population, intervention, study design, and evidential strength. A useful representation system therefore needs an explicit account of which relation is being optimized, at which level, and for which consumer.

This article asks: how can general embeddings be reorganized into representations that are more expressive, task-specific, and auditable without unnecessarily retraining the entire encoder? Our answer is layered semantic refinement (LSR): a separation-of-responsibilities architecture in which a base encoder supplies transferable semantics, while one or more explicit refinement modules encode task, relational, hierarchical, or governance requirements. The base representation remains accessible as a fallback and as a control for evaluating information loss.

The contribution is a framework-level synthesis and methodological protocol rather than an empirical claim or a claim that each component is new. Specifically, the article: (i) defines semantic organization in operational terms; (ii) distinguishes the token-to-embedding pipeline and the different meanings of prompt embedding and prompt token; (iii) integrates projection-based, graph-based, hierarchical, rule-based, clustering, multi-view, late-interaction, and prompt-conditioned mechanisms within a common formalism; (iv) positions LSR against direct precedents in multi-view document representation, ColBERT-style retrieval, instruction-conditioned embedding, Matryoshka representations, and structure-aware document encoding; (v) specifies one canonical, frozen-encoder LSR implementation; (vi) defines a token-aware consumption protocol that separates embedding-time savings from generation-time savings; (vii) specifies losses, diagnostics, ablations, and statistical tests required to attribute improvement to refinement; (viii) provides a corrected experimental blueprint in which validation, not the test set, controls model selection; and (ix) translates the framework into deployment and governance requirements for production vector systems. The central claim is therefore integrative: LSR offers a common separation-of-responsibilities language and an auditable evaluation contract for systems whose individual mechanisms already have important precedents.

The phrase “layered semantic refinement” should not be read as a claim that the present article invented projections, graph neural networks, multi-view retrieval, prompt tuning, or hierarchical embeddings. Instead, LSR names an integration pattern: a base representation is made conditional, relational, or multi-scale by explicit refinement modules, and the resulting system is evaluated for utility, attribution, preservation, and operational compatibility.

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

2. Conceptual and Technical Foundations

2.1 Tokens, contextual states, and pooled embeddings

A token is a discrete unit produced by a tokenizer and may correspond to a word, subword, punctuation mark, byte sequence, or special symbol. A token is therefore an index or symbol in a model vocabulary; it is not yet a semantic vector. The embedding lookup maps that token identity to an input vector e_i∈R^d. After positional information and repeated self-attention layers are applied, the model produces a contextual state h_i^(ℓ) for token i at layer ℓ. These levels must be kept distinct: tokenization determines the discrete sequence and its token budget; token embeddings are the initial continuous inputs; contextual states are position-specific, context-dependent representations; and a pooled text embedding is a fixed-length vector obtained by aggregating contextual states for retrieval or downstream prediction. Tokenization controls the granularity at which a model receives linguistic input and is therefore part of the representation system rather than a neutral preprocessing detail. Excessive fragmentation of biomedical names, chemical formulas, legal citations, or morphologically rich words can alter computational cost and the accessibility of domain information.

Let $x=(t_1,\ldots,t_n)$ denote a token sequence, let E(t_i)=e_i be its input token embedding, and let H^(ℓ)=(h_1^(ℓ),…,h_n^(ℓ)) be the contextual states at transformer layer ℓ. A transformer encoder applies a contextual operator to the sequence of token embeddings, positional signals, and attention masks, so that h_i^(ℓ) depends on neighboring tokens and on the layer depth. A fixed-length sentence or document representation h may then be obtained from a designated token, mean pooling, weighted pooling, attention pooling, or a learned aggregation function. These alternatives are not interchangeable: pooling determines which token-level information is retained, how sequence length affects the vector, and whether similarity is sensitive to salient but infrequent terms. In compact notation, the pipeline is t_i → e_i=E(t_i) → h_i^(L)=F_L(e_1,…,e_n) → h=Pool(H^(L)) → z=g_φ(h,S). The final embedding z is consequently several transformations removed from the discrete token sequence.

The distinction is operationally important for token economy. A text encoder consumes a sequence of tokenizer-produced positions, whereas a vector index stores a fixed-length embedding after encoding. Storing one vector per document does not mean that the document required only one token: the full token sequence was consumed during encoding,

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

usually offline. Conversely, a shorter runtime prompt can reduce context length for a query or generator, but it does not remove the token cost of the original document unless the document is pre-encoded and no longer reinserted into the model context.

2.2 From BERT to sentence embeddings

BERT established bidirectional transformer pretraining with masked language modeling and a sentence-level auxiliary objective in its original formulation (Devlin et al., 2019). Its contextual states are highly useful for downstream fine-tuning, but raw pooled BERT vectors are not automatically calibrated so that cosine distance reflects sentence-level semantic similarity. Sentence-BERT addresses the computational and geometric mismatch through Siamese or triplet architectures that encode each sentence independently and optimize simple vector comparisons (Reimers & Gurevych, 2019). SimCSE further demonstrates that contrastive objectives can substantially improve sentence embedding geometry; its unsupervised variant treats dropout-corrupted views of the same sentence as positives, whereas the supervised variant uses natural-language-inference relations (Gao et al., 2021).

These developments clarify an important distinction. An encoder can contain information that is not linearly or metrically accessible under the downstream similarity function. Refinement can therefore mean at least three different operations: extracting information already latent in h, injecting external structure not present in h, or trading general information for improved task discrimination. A sound evaluation must identify which of these claims is being made.

2.3 Geometry: similarity, anisotropy, hubness, and collapse

For L2-normalized vectors u and v, the inner product equals cosine similarity. This convenient equivalence does not guarantee a well-behaved space. Contextual embeddings may be anisotropic, occupying a narrow cone in which unrelated items exhibit high similarity. In high-dimensional nearest-neighbor systems, hubness can cause a small number of points to appear disproportionately often in neighbor lists. Refinement objectives may also induce representational collapse, mapping diverse inputs into a small region because class separation is rewarded more strongly than within-class structure.

$$\cos(u,v) = (u^{T}v)/(\|u\|_2\|v\|_2); \quad \bar{u} = u/\|u\|_2$$

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

Consequently, a two-dimensional t-SNE or UMAP plot cannot establish semantic organization. t-SNE emphasizes local neighborhoods and may distort global separation; UMAP depends on neighborhood size, minimum distance, metric, initialization, and sampling. Visualizations are valuable for hypothesis generation but must be accompanied by original-space diagnostics: distributions of norms and cosines, intra- versus inter-group similarity, k-nearest-neighbor hubness, neighborhood stability, calibration, and performance under controlled perturbations.

2.4 Hierarchy, structure, and multiple semantic views

The adjective hierarchical can refer to different objects: compositional units (tokens → sentences → sections → documents), labels (disease → subtype), tasks (lexical prediction → entity recognition → relation extraction), clusters (documents → subtopics → themes), or dimensional prefixes in Matryoshka representations. These structures share an intuition of nested resolution but require different architectures and evaluation criteria. A tree is not always appropriate: documents may belong to several topics, emotions may co-occur, and institutional taxonomies may change. Directed acyclic graphs, soft memberships, and multiple views can be more faithful.

2.5 Prompt embeddings: terminology and representational mechanisms

The expression prompt embedding is used inconsistently and should be disambiguated before it is related to organized embeddings. A **prompt token** can be a discrete vocabulary token in a textual instruction, such as “retrieve”, “evidence”, or “stance”, or a virtual token represented only by a learned continuous vector. A **prompt embedding** can therefore mean either the vector E(p_j) associated with a prompt token, a pooled vector e(I) representing an instruction, or the complete matrix P of learned virtual prompt vectors. These objects are not interchangeable. A discrete instruction adds tokenizer-visible positions; a soft prompt supplies learned positions without a human-readable lexical identity; and an instruction-conditioned text embedding is the final representation of the target text under a declared semantic criterion. In this article, prompt token denotes an input position or virtual slot, prompt embedding denotes its continuous representation or the embedding of the instruction, and text embedding denotes the representation consumed by retrieval, classification, clustering, or generation.

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

These mechanisms operate at different locations. A hard prompt modifies the observable token sequence. Prompt tuning learns a matrix P∈R^(m×d) of m virtual token embeddings and prepends it to the embedded input while the language-model parameters remain frozen (Lester et al., 2021). Prefix tuning extends the intervention by learning continuous activations that influence attention across transformer layers, particularly through prefix key–value states (Li & Liang, 2021). P-Tuning v2 places trainable prompts at multiple layers and has been evaluated as a deep prompt-tuning strategy for natural-language understanding (Liu et al., 2022). Instruction-aware embedding models such as INSTRUCTOR instead learn an encoder F that maps an instruction I and text x to an embedding whose geometry is conditioned on the declared task and domain (Su et al., 2023). Multi-view retrieval and late-interaction models provide complementary alternatives in which several document or token vectors are retained rather than collapsed into a single vector (Zhang et al., 2022; Santhanam et al., 2022).

$$P = [p_1,\ldots,p_m] \in R^{(m\times d)}; \quad e(I)=Pool(E(I)); \quad h(I,x)=F([I;x]); \quad z=g_\varphi(h(I,x),S).$$

The formulas expose three distinct control points. P changes the internal computation by supplying learned continuous context; e(I) is a reusable vector representation of an instruction or relation; and I changes the semantic criterion expressed to an instruction-trained encoder. A soft prompt can implement a parameter-efficient, model-internal refinement layer; an instruction can select a task-conditional base geometry before post-encoding refinement; and an instruction embedding can serve as a routing or conditioning signal when the model supports that interface. The combined architecture therefore supports prompt → encoder → refinement, while retaining the unprompted base vector as a control. This arrangement is especially relevant when the same corpus must support several notions of similarity, such as topic, stance, evidential role, or writing style.

Prompt conditioning does not automatically yield interpretability. Soft prompt vectors are continuous parameters whose nearest vocabulary tokens rarely provide a faithful explanation of their function. Discrete instructions are readable, but the model may respond to wording, order, verbosity, or spurious lexical cues. Consequently, prompt-conditioned embedding systems require paraphrase invariance tests, prompt-template ensembles or sensitivity analyses, unseen-instruction evaluation, and controls in which semantically irrelevant instructions are substituted. They also require an explicit token accounting rule: a shorter prompt is useful only if it preserves the target relation and does not cause the model

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

to re-encode a larger document or to retrieve so many candidates that the downstream context becomes longer.

Token and representation terminology

The largest production savings usually come from a pipeline rather than from the prompt alone: encode documents offline; cache base and relation-specific embeddings; use a short prompt token or a relation identifier to route the query; retrieve a small candidate set; expand only validated evidence; and send a compact, deduplicated context to the generator. A prompt embedding can reduce repeated textual instructions and enable reuse of a relation-specific cache, but it cannot be assumed to be compatible with unprompted corpus vectors. If the instruction changes document-side geometry, documents must be re-embedded or the prompt must be restricted to routing and reranking. Token savings should therefore be reported separately for tokenizer positions, encoder FLOPs, vector-storage bytes, retrieved-context tokens, and generated tokens.

Let L_x be the number of tokenizer-visible positions in the target text and L_I the number of positions in a textual instruction. A hard-prompt encoder consumes approximately $L_{hard}=L_I+L_x$ positions. A soft prompt with m virtual prompt slots consumes $L_{soft}=m+L_x$ positions. If $m \ll L_I$, the model can avoid repeatedly spelling out a long instruction or a set of demonstrations, reducing context length and self-attention work at the embedding stage. For a standard full-attention Transformer, the dominant attention cost is approximately quadratic in sequence length, so reducing repeated instruction positions can be valuable. However, virtual slots still occupy positions, consume memory and compute, and do not automatically reduce the tokens in the underlying document or the tokens generated by a downstream language model.

Token economy and the limits of soft prompts

Object	Definition	What it consumes or stores
Token	A tokenizer-produced discrete symbol or vocabulary index.	Consumes one tokenizer position in the model input.

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

Token embedding	The initial continuous vector E(t_i) associated with a token identity.	Occupies one vector at the input interface; it is not yet context-dependent.
Contextual token state	A position-specific vector h_i^(ℓ) after transformer layers and attention.	Depends on neighboring tokens and layer depth; many states are produced per input.
Text embedding	A pooled or aggregated fixed-length vector used by a downstream consumer.	Stored or indexed after encoding; it does not reveal the original token count.
Prompt token / prompt embedding	A discrete instruction token or a learned virtual slot and its continuous vector.	Adds tokenizer-visible positions or virtual slots; may replace repeated text but still consumes compute.

Table 1. Recurrent failure modes in general-purpose embedding pipelines.

Failure mode	Technical manifestation	Operational consequence	Required diagnostic
Relation superposition	Topic, stance, sentiment, and role share one vector/metric	Queries retrieve the wrong kind of similarity	Task-specific probes and paired contrast sets
Anisotropy and hubs	Vectors occupy a narrow cone; a few points dominate k-NN lists	Biased retrieval and unstable ranking	Cosine distribution, isotropy, and k-occurrence curves
Domain shift	Terminology and relations differ from pretraining data	Performance degrades on new sources or periods	Temporal, source, geographic, and vocabulary splits

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

Failure mode	Technical manifestation	Operational consequence	Required diagnostic
Structural erasure	Flattening removes section, column, citation, or parent–child relations	Semantically plausible but contextually incoherent results	Structure-aware controls and edge ablations
Projection overinterpretation	Two-dimensional maps create apparently separated islands	Unsupported causal or semantic claims	Multiple seeds plus metrics in the original space
Refinement collapse	Task loss compresses legitimate within-class variation	Poor transfer and brittle nearest neighbors	Transfer probes, rank, covariance, and neighbor diversity

3. Layered Semantic Refinement: Framework, Scope, and Formalization

3.1 Architectural definition

Let $f\theta$ map an input x to a base embedding $h=f\theta(x)\in R^d$. A refinement module $g\varphi$ receives h and, when applicable, contextual structure S such as graph edges, section membership, rules, metadata, or a taxonomy. It produces one or more organized views $z=(z^1,\ldots,z^m)$:

$$h = f\theta(x), \quad z^j = g\varphi_j(h,S), \quad \hat{y} = q\omega(z^1,\ldots,z^m).$$

The encoder θ may be frozen, partially tuned, or jointly optimized. Freezing is the cleanest first experiment because it isolates the capacity of the refinement layer and limits catastrophic forgetting. The organized views need not replace h. Retaining both h and z supports fallback retrieval, transfer tests, ensemble decisions, and rollback. Each deployed vector must be accompanied by a schema that records encoder identity, refinement identity, dimension, normalization, taxonomy version, training data window, and intended similarity metric.

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

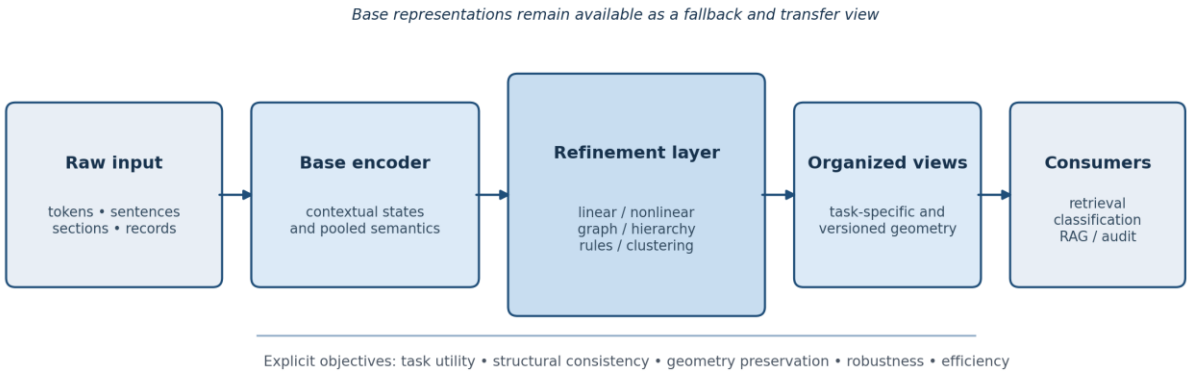

Figure 1. Layered semantic refinement separates general semantic encoding, explicit organization mechanisms, and downstream consumers.

3.2 What counts as semantic organization?

A representation is not organized merely because vectors have been transformed, compressed, or visualized. Under LSR, organization is a falsifiable property defined relative to an intended relation R. Four conditions are necessary: (1) R is operationally specified through labels, edges, constraints, or evaluation pairs; (2) a metric demonstrates that R is more accessible or more consistently represented after refinement; (3) the effect survives resampling, seed variation, and relevant distribution shifts; and (4) ablation shows that the proposed structural information—not only extra parameters or test-set tuning—accounts for the improvement.

3.3 Multi-objective learning

Because task specialization can erase transferable information, a refinement objective should normally combine utility with preservation and regularization. A generic objective is:

$$L = \lambda task\ Ltask + \lambda struct\ Lstruct + \lambda pres\ Lpres + \lambda fair\ Lfair + \lambda reg\ ||\varphi||^2.$$

Ltask may be cross-entropy, ranking, or contrastive loss. Lstruct encodes graph edges, parent–child consistency, section composition, or rule satisfaction. Lpres protects useful geometry, for example by matching pairwise similarities or distilling the base encoder. Lfair constrains disparities when the application warrants group-level assessment. The λ

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

coefficients express policy and scientific assumptions; they must be selected on validation data and reported, not treated as incidental hyperparameters.

3.4 Principal refinement families

Table 2. Taxonomy of refinement mechanisms and their inductive biases.

Family	Canonical form	Best suited to	Primary risk
Linear projection	$z=Wh+b$	Low-resource adaptation, compression, separability testing	Mistaking classifier capacity for representational improvement
Nonlinear adapter	$z=W_2\sigma(W_1h+b_1)+b_2$	Nonlinear task boundaries and controlled specialization	Overfitting and geometry collapse
Graph refinement	$z_i=\varphi(h_i,AGG_j\psi(h_i,h_j,e_{ij}))$	Tables, citations, entities, social or knowledge relations	Noisy edges, leakage, and expensive coupled inference
Hierarchical aggregation	sentence→section →document; child→parent	Semi-structured documents and nested labels	Rigid or incorrect hierarchy
Rule-augmented view	$z=[h;r(x)]$ or constrained score fusion	Auditable institutional or scientific constraints	Proxy discrimination and brittle rules
Clustering/index layer	tree, forest, prototypes, or region IDs over h	Exploration, routing, and label-scarce corpora	Unstable clusters and overinterpreted labels
Multi-view/mixture	$z=\{z_{topic},z_{stance},z_{time},\ldots\}$	Queries requiring different semantic relations	Routing inconsistency and higher maintenance cost

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

Family	Canonical form	Best suited to	Primary risk
Prompt-conditioned	h(I,x) or virtual tokens P	Task-adaptive geometry without full encoder retraining	Prompt sensitivity, opaque soft prompts, and version drift

A token-aware deployment variant prepends a small learned prompt bank P_topic, P_evidence to the query encoder, with m reported explicitly. Document vectors are either encoded with the same relation condition or kept generic when the prompt is used only for routing. The deployment contract records the prompt placement, tokenizer revision, model revision, pooling rule, normalization, index compatibility, and whether corpus documents were re-embedded. This condition is necessary because prompt-conditioned query vectors and generic document vectors are not automatically in the same metric space.

The canonical objective is $L=\lambda_{task}(L_{topic}+L_{evidence})+\lambda_{pres} L_{pres}+\lambda_{div} L_{div}+\lambda_{reg}||\varphi||^2$. L_task trains the declared relation, L_pres distills selected pairwise similarities from h, and L_div prevents topic and evidence views from collapsing into identical neighborhoods. The primary comparison is against a parameter-matched MLP applied directly to h, followed by instruction-conditioned and multi-vector baselines. The base vector is never discarded: it remains available for generic retrieval, transfer probes, rollback, and measuring information loss. This narrow instantiation makes it possible to test whether the general LSR separation of responsibilities is useful without claiming that every listed refinement family has been validated.

To make the framework experimentally falsifiable, we define one canonical instantiation rather than treating every mechanism in the taxonomy as part of a single model. The canonical system freezes a sentence encoder f_θ, uses mean pooling followed by L2 normalization to obtain h, and learns two parameter-matched relation-specific adapters: g_topic and g_evidence. Each adapter maps h to k dimensions, applies LayerNorm and GELU, and returns a normalized vector. The system retains h as a generic view, produces z_topic and z_evidence as specialized views, and uses a lightweight router only when the query explicitly declares the relation. If observable document structure is available, a graph module may be added to the evidence view; otherwise the graph channel is absent rather than inferred from arbitrary proximity.

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

3.5 Canonical LSR instantiation

4. Evidence from Related Architectures

4.1 TabEmb: semantic encoding followed by relational refinement

TabEmb explicitly decouples semantic column encoding from structural table modeling (Hoseinzade et al., 2026). An LLM produces a representation for each column, and a graph module injects inter-column information before column type annotation, column property annotation, and table type annotation. The architecture directly supports the LSR thesis: a general semantic representation is retained, while a lightweight structure-aware module models relations that sequence linearization can obscure. The authors report average gains over strong baselines across the three annotation tasks; these results must be described per task and comparison rather than collapsed into a universal percentage. The study also illustrates a necessary control: a graph should outperform a parameter-matched non-graph module or randomized-edge graph before gains are attributed to relational structure.

4.2 TAROT: multi-granular representations for person–job fit

TAROT models semi-structured profiles and job descriptions across sentence, section, individual, and interaction levels (Cao et al., 2024). Attention-based aggregation and multi-task co-pretraining constrain the information learned at each level, while cross-attention represents compatibility as a relation rather than an isolated person or job attribute. The paper reports relative improvements in NDCG@5 in proprietary LinkedIn settings. The evidence is operationally relevant but not independently reproducible with the same data; therefore, it supports the architectural principle more strongly than a universally portable effect size.

4.3 TopicForest: post-hoc organization for navigable literature

TopicForest constructs an embedding-driven hierarchical clustering and labeling interface for biomedical literature (Chang et al., 2025). Unlike a supervised projection, it does not necessarily modify the encoder. Instead, it organizes documents into a multi-scale navigable structure over the existing metric space. This is a legitimate refinement layer because the output is not merely a scatter plot: it is a queryable hierarchy with labels and traversal operations. Nevertheless, every dendrogram imposes a hierarchy, even when the underlying topics overlap. Stability under resampling, sensitivity to linkage and distance,



Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

expert assessment, and allowance for multi-topic membership are therefore central to validity.

4.4 Hierarchical multitask and Matryoshka representations

Sanh, Wolf, and Ruder (2019) show that semantically ordered auxiliary tasks can be supervised at different encoder depths, linking task hierarchy to representation depth. Matryoshka representation learning, in contrast, optimizes nested prefixes of a vector so that different dimensional budgets remain useful (Kusupati et al., 2022). Hanley and Durumeric (2025) combine multilingual Matryoshka embeddings with level-wise clustering to identify stories, narratives, and themes at different granularity. These methods should not be conflated: one organizes supervision across network depth, another across embedding dimensionality, and the third uses nested granularity in a hierarchical clustering pipeline.

4.5 Prompt tuning as parameter-efficient internal refinement

Prompt tuning provides a particularly direct relation between prompt embeddings and LSR. Instead of updating θ, it learns a comparatively small prompt matrix P while the pretrained model remains frozen (Lester et al., 2021). This is structurally analogous to a refinement adapter, but its intervention occurs before or inside the encoder rather than after the final vector. Prefix tuning and deep prompt tuning further distribute continuous conditioning across transformer layers (Li & Liang, 2021; Liu et al., 2022). The resulting mechanism can steer which latent features are amplified, suppressed, or made accessible to the pooling operation.

From an LSR perspective, a bank of soft prompts {Ptask, Pdomain, Prelation} can be treated as a set of reusable refinement experts. A router selects one prompt or composes several prompts before encoding. This offers low per-task storage and can preserve a common frozen backbone, but it also creates new experimental requirements: compare against a parameter-matched post-encoder adapter; report sensitivity to prompt length and initialization; test transfer across model revisions; and determine whether combining prompts is approximately compositional or produces interference. A prompt learned for sentiment should not be assumed to preserve topical or retrieval geometry.

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

4.6 Instruction-conditioned embeddings as query-dependent organization

Instruction-conditioned embedding models move the refinement criterion into natural language. INSTRUCTOR represents each input together with an instruction specifying the task and domain and can produce different vectors for the same text under different instructions (Su et al., 2023). This property directly addresses the functional flatness problem: instead of forcing one geometry to serve all relations, the instruction identifies the relation that should govern proximity. Later instruction-following embedding research similarly treats instructions as controls over what information the vector should emphasize (Peng et al., 2024; Wang et al., 2024).

This approach can be interpreted as conditional metric learning. Let I_topic request similarity by subject, while I_stance requests similarity by position toward a proposition. For the same x, h(I_topic,x) and h(I_stance,x) need not have identical neighborhoods. The relevant test is not whether the two vectors differ, but whether each induced neighborhood better satisfies its declared relation. Evaluation therefore requires crossed conditions: each prompt-conditioned representation must be tested on both its target relation and non-target relations, with semantically equivalent prompt paraphrases and deliberately misleading instructions.

Prompt-conditioned embeddings and post-hoc refinement are complementary. Instructions offer flexible, query-time control but may require re-encoding documents if the instruction is applied symmetrically to queries and corpus items. Post-encoder refinement can be cached and indexed efficiently but is less flexible when the relation changes. A practical hybrid can maintain a generic corpus index, apply an instruction-aware query encoder for routing, and use relation-specific refiners or rerankers on a restricted candidate set. Whether asymmetric conditioning is valid must be established empirically because query and document prompts can introduce incompatible geometries.

The literature supports the individual mechanisms used by LSR. Multi-view retrieval, instruction-conditioned embeddings, Matryoshka prefixes, hierarchical document encoders, graph-based semantic–structural embeddings, and hierarchical clustering are all established directions. The defensible contribution of LSR is the integration pattern and the evaluation contract: an encoder-base is kept as a control; a declared relation selects or trains a refinement; preservation and structural attribution are measured; and the complete

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

encoder–refiner–taxonomy–index contract is versioned. The framework should not be described as inventing any one of these mechanisms.

4.8 Scope and originality boundary

ColBERTv2 (Santhanam et al., 2022) takes a different route: it retains multiple contextual vectors at token granularity and computes relevance through late interaction. Its residual compression reduces the storage cost of a multi-vector index, but the model still makes an explicit quality–memory trade-off. LSR should therefore distinguish multi-view embeddings, multi-vector token representations, and post-hoc hierarchical indices. They all overcome a single-vector bottleneck, but they do so at different locations and with different operational costs.

The single-vector bottleneck is not only a motivation introduced by LSR; it is an established design problem in dense retrieval. Zhang et al. (2022) learn multiple document embeddings through independent viewers and use a global-local loss with annealed temperature to prevent the views from collapsing. The work is a direct precedent for the LSR claim that one document may need several operationally distinct views, although its scope is open-domain retrieval rather than a general framework.

4.7 Multi-view document representations and late interaction

Table 3. Comparative interpretation of central studies.

Study	Base unit	Added structure	Evidence type	Transferable lesson
Sanh et al. (2019)	Shared encoder states	Semantic ordering of tasks across layers	Public NLP benchmarks	Task hierarchy can guide where supervision enters a network
TAROT (2024)	Sentence and section	Sentence→section →individual→interaction	Proprietary person–job data	Represent the unit and relation at which decisions occur

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

Study	Base unit	Added structure	Evidence type	Transferable lesson
TopicForest (2025)	Biomedical article	Hierarchical clusters and generated labels	Biomedical corpus and user-facing structure	Post-hoc organization can make a vector corpus navigable
Hanley & Durumeric (2025)	News article	Nested embedding granularity plus hierarchy	Public similarity benchmark and real-world corpora	Granularity can be encoded and queried at multiple levels
TabEmb (2026)	Table column	Graph over inter-column relations	Public table-annotation tasks	Separate semantic encoding from observable relational structure
Multi-View Retrieval (2022)	Document	Multiple viewers and global-local loss	Public retrieval benchmarks	One document may require several views
ColBERTv2 (2022)	Token	Multi-vector late interaction and compression	Public retrieval benchmarks	Fine-grained matching trades memory for quality
INSTRUCTOR (2023)	Text + instruction	Instruction-conditioned geometry	70 public embedding tasks	The relation can be declared at input time
HiStruct+ (2022)	Sentence and section	Section titles and hierarchical position	Three public summarization datasets	Observable document structure can be

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

Study	Base unit	Added structure	Evidence type	Transferable lesson
				injected and ablated
SLED (2025)	Transformer layers	Spectral semantic diffusion and normalization	Preprint experiments	Layered semantic diffusion is a close but distinct mechanism

5. Experimental Blueprint

5.1 Research questions and hypotheses

A publishable evaluation should separate predictive performance, representational quality, structural attribution, and operational cost. The following research questions prevent a single accuracy number from carrying claims it cannot support.

1. RQ1—Task utility: Does refinement improve held-out performance relative to parameter-matched heads operating directly on the base embedding?
2. RQ2—Representation quality: Does the refined vector improve linear probes, k-nearest-neighbor classification, retrieval, calibration, and neighborhood consistency without relying on a more powerful downstream classifier?
3. RQ3—Structural attribution: Do gains decrease when edges, hierarchy levels, rules, or labels are randomized or removed?
4. RQ4—Preservation: Which general capabilities and local neighborhoods are lost after specialization?
5. RQ5—Robustness and fairness: Are effects stable across seeds, time, source, vocabulary shift, perturbations, and relevant subgroups?
6. RQ6—Prompt controllability: Does a prompt or instruction alter neighborhoods in the intended direction while remaining stable across semantically equivalent formulations?

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

7. RQ7—Efficiency: What quality–latency–memory frontier is achieved when encoder, prompt conditioning, refinement, indexing, and reindexing costs are measured separately?

5.2 Demonstration dataset and split discipline

AG News contains four topical classes—World, Sports, Business, and Sci/Tech—and is appropriate as a transparent demonstration benchmark (Zhang et al., 2015). It is a flat topical dataset and therefore cannot, by itself, validate hierarchical consistency, parent–child constraints, or multi-relation control. In the revised protocol, AG News is used only for the flat task-utility demonstration. The principal LSR experiment must use a dataset in which the same items carry at least two relations, for example topic plus stance, topic plus evidential role, or story similarity at multiple granularities. If no public dataset provides all relations, the study should publish a small, documented annotation layer and report inter-annotator agreement. Hierarchical claims require an explicit taxonomy or graph and should not be inferred from the four AG News labels.

A compact sentence encoder such as all-MiniLM-L6-v2 can generate base vectors, but the exact model revision, tokenizer revision, pooling, truncation length, normalization, batch size, and library versions must be recorded. The token budget must also be reported: mean and percentile document length, mean and percentile query length, prompt-token count m or instruction length L_I, encoder sequence length, and the number of retrieved context tokens ultimately sent to a generator. For multilingual claims, AG News alone is insufficient; equivalent corpora and aligned semantic pairs across languages are required.

5.3 Baselines and ablations

- Base embedding + linear classifier: tests linearly accessible task information.
- Base embedding + parameter-matched multilayer perceptron: controls for added nonlinear capacity.
- Refined embedding + frozen linear probe: tests whether z itself is more usable, rather than only the joint model.
- End-to-end fine-tuned encoder: establishes the quality ceiling and cost trade-off.
- Randomized structure: shuffles graph edges, hierarchy membership, or rule assignments while preserving parameter count.

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

- Preservation-loss ablation: quantifies the trade-off between specialization and general geometry.
- Dimension-matched random projection and PCA: distinguish learned organization from compression alone.

For prompt embeddings, additional baselines are required: an unprompted encoder; a fixed discrete instruction; paraphrases of the same instruction; a semantically irrelevant instruction of similar length; randomly initialized frozen virtual tokens; trained soft prompts with matched parameter counts; a post-encoder adapter; an INSTRUCTOR-style instruction-conditioned encoder; and an external instruction embedding used only for routing. Crossed evaluation should compare I_topic, I_stance, and I_evidence on datasets annotated for all three relations. Prompt robustness should report mean performance and dispersion across templates rather than selecting the best wording on the test set. For the vector-consumption claim, the study should add a multi-view document baseline and a ColBERT-style multi-vector or late-interaction baseline where retrieval is the target task.

Every experiment must report two separate budgets. The representation budget counts tokenizer positions and transformer work used to encode the query or document. The consumption budget counts the text tokens delivered to the downstream generator or reranker. A soft prompt can reduce the first budget when a short bank of m virtual tokens replaces a long repeated instruction, but it does not reduce the tokens of x. Offline document encoding can remove document encoding from the runtime path, while coarse-to-fine routing and evidence expansion can reduce the consumption budget. The paper must not call these savings “token reduction” without specifying which budget was reduced.

5.3.1 Explicit token and prompt accounting

5.4 Minimal implementation with correct validation

The following pseudocode corrects a frequent methodological error in compact examples: early stopping must use validation data, never the official test set. The base embeddings are precomputed, the encoder is frozen, and the best checkpoint is selected by validation macro-F1.

```
# Pseudocode: frozen encoder, validation-controlled refinement
encoder = SentenceTransformer(MODEL_REVISION)
H_train, H_val, H_test = encode_and_cache(train, val, test)
```

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

```
model = Sequential(
    Linear(d, k), LayerNorm(k), GELU(), Dropout(0.10)
)
classifier = Linear(k, n_classes)

for seed in SEEDS:
    initialize(seed)
    for epoch in range(MAX_EPOCHS):
        z = normalize(model(H_train_batch))
        loss = cross_entropy(classifier(z), y_batch)
        loss += lambda_pres * preserve_pairwise_similarity(H_train_batch, z)
        optimize(loss)
if macro_f1(y_val, predict(normalize(model(H_val)))) improves:
            save_checkpoint()
        elif patience_exhausted:
            break
    load_best_checkpoint()
    evaluate_once(H_test)
```

A minimal runtime pattern is: tokenize the query once; select a relation-specific prompt token bank or short instruction; compute a query vector; retrieve a small candidate set; apply structure-aware reranking; compress and deduplicate evidence; and send only the final context to the generator. The embedding vector itself is not a textual token and cannot be counted as a generated-token saving. The saving arises from avoiding repeated instructions, avoiding full-document re-encoding, and reducing the amount of retrieved text that reaches the consumer.

Efficiency should be measured as a quality–latency–memory–token frontier. Report encoder-only latency, refinement latency, index lookup latency, reindexing cost, vector-storage bytes, mean and p95 tokenizer positions, retrieved-context tokens, and generated tokens. For prompt-conditioned systems, report the number of textual instruction tokens L_I,

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

the number of virtual prompt slots m, whether the instruction is cached, whether documents are re-embedded, and whether the prompt is applied symmetrically to queries and documents. A valid token-saving claim should compare equal-quality systems under the same task metric and should specify whether the saving is in embedding-time context, runtime context, generation, or storage.

5.5 Token-aware consumption and efficiency

5.6 Metrics and statistical inference

Classification should report accuracy, macro-F1, class-wise precision/recall/F1, calibration error, and confusion matrices. Retrieval should report recall@k, mean reciprocal rank, and NDCG@k using relevance judgments aligned with the intended relation. Hierarchical tasks require hierarchical precision/recall, ancestor consistency, and path-sensitive error costs. Geometry diagnostics should include alignment and uniformity, intra-/inter-group similarity, neighborhood overlap between h and z, hubness, effective rank, and covariance spectra.

Optimization variability and sampling uncertainty are distinct. Multiple independently initialized runs estimate seed sensitivity; stratified bootstrap intervals over held-out examples estimate sampling uncertainty. McNemar's test evaluates paired disagreement between two classifiers on the same examples. When numerous variants are compared, multiplicity control and a preregistered primary outcome reduce selective reporting. Effect sizes and confidence intervals are more informative than isolated p-values.

Table 4. Reporting template for an empirical LSR study; no values are claimed in the present article.

Condition	Task metric (mean±SD)	95% CI	p50/p95 latency	Memory	Structural ablation
Base + linear head	To be measured	To be measured	To be measured	To be measured	Not applicable
Base + matched MLP	To be measured	To be measured	To be measured	To be measured	Not applicable

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

Condition	Task metric (mean±SD)	95% CI	p50/p95 latency	Memory	Structural ablation
LSR projection	To be measured	To be measured	To be measured	To be measured	Remove preservation loss
LSR graph/hierarchy	To be measured	To be measured	To be measured	To be measured	Randomize edges/levels
End-to-end fine-tuning	To be measured	To be measured	To be measured	To be measured	Freeze encoder

Important: the 85–91% accuracy and 8–25 ms values in the source manuscript were hypothetical design values. They have been removed from the results narrative because they were not generated by an executed experiment. They may be used only in a clearly labeled simulation or preregistered planning section, never as evidence for the proposed method.

6. Application Patterns

6.1 Structure-aware retrieval and RAG

Conventional RAG pipelines often index isolated chunks and discard document identity, section role, version, citation relations, and evidential dependencies. An LSR implementation can maintain coordinated views for chunk content, section context, document-level topic, and relation edges. Retrieval then proceeds coarse-to-fine: route to a domain or document family, rank sections, retrieve evidence-bearing chunks, and expand context along validated relations. The retriever's evidence recall must be evaluated separately from the generator's factuality; otherwise, generation quality can conceal retrieval failure (Lewis et al., 2020).

Prompt embeddings add a task-control layer to this pipeline. A retrieval instruction can specify whether the query seeks definitional equivalence, supporting evidence,

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

contradiction, methodological similarity, or a clinically comparable population. The same query text may therefore generate different vectors and candidate sets. Soft prompts can provide a parameter-efficient domain adapter for the query encoder or reranker, while discrete instructions permit runtime control. The token-economy benefit is obtained when the instruction is short, cached, or represented by m virtual prompt slots rather than repeated demonstrations and long textual policies. This benefit is limited to the embedding or routing stage unless the resulting candidate set is also smaller; the downstream generator still consumes the retrieved text. Corpus-side re-encoding is unnecessary only when the model has been trained for asymmetric query/document instructions or when prompt conditioning is restricted to routing and reranking. Otherwise, mixing prompted queries with unprompted document vectors may violate metric compatibility.

6.2 Prompt libraries as semantic interfaces

A production system can maintain a governed prompt library in which each prompt is treated as a versioned semantic interface. Each entry should specify intended relation, supported domains, model and tokenizer revision, prompt length or text, number of virtual prompt slots, pooling, normalization, evaluation datasets, known failure modes, token budget, and compatibility with existing indexes. Soft prompts require binary artifact hashes and model-layer compatibility; natural-language instructions require exact text, language, and template variables. Prompt versioning is therefore as important as encoder and index versioning. A prompt library can reduce runtime token consumption when a short relation identifier or learned prompt replaces a long repeated instruction, but the registry must record whether this is an actual reduction in tokenizer positions, a reduction in retrieved context, or only a reduction in human-authored prompt text.

Prompt selection may be manual, rule-based, or learned. A learned router introduces another model whose mistakes must be observable. Logging should record the selected prompt, confidence, alternative prompts considered, prompt-token count, instruction-token count, and downstream effect. In high-stakes applications, unrestricted user-provided instructions should not silently redefine the retrieval relation; approved prompt schemas, input validation, and fallback to a generic embedding reduce prompt-injection and governance risks. Caching prompt embeddings or prompt key–value states is useful only when the underlying model and attention implementation support reuse; it should not be conflated with caching the final document embedding.

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

6.3 Table and semi-structured data understanding

Tables, forms, résumés, clinical notes, and regulatory documents combine text with stable structural roles. Encoding every field independently loses relations, while linearizing the entire object may exceed context windows and obscure two-dimensional or hierarchical organization. A semantic encoder followed by graph or hierarchical refinement provides a modular alternative. Edge types should correspond to observable relations such as same row, foreign key, section membership, temporal succession, or citation—not arbitrary proximity.

6.4 Multi-faceted recommendation

A user's preference is neither unitary nor stationary. Separate views can represent short-term intent, long-term interest, location, seniority, novelty tolerance, and policy constraints. Items receive compatible views, and a query-dependent router determines which geometry applies. This design can improve controllability, but it can also intensify feedback loops by narrowing exposure. Offline relevance must therefore be paired with exposure, diversity, calibration, and counterfactual evaluation.

6.5 Scientific and clinical knowledge organization

Scientific documents require more than topical similarity. Study design, population, intervention, comparator, outcome, direction of effect, and evidence quality are distinct axes. LSR can combine text embeddings with explicitly extracted methodological attributes and citation relations. In high-stakes settings, however, the refined vector must not be treated as a clinical conclusion. Provenance, uncertainty, human review, and rejection of out-of-scope queries are mandatory system properties.

6.6 Sentiment, stance, and emotion

Topic and affect often conflict within a single space. A sentence may discuss the same product while expressing opposite sentiment; stance depends on a target; emotion can be multilabel and causally attributed. Multi-view refinement permits independent topic, sentiment, target, and causal-role representations. Evaluation should include negation, irony, implicit sentiment, cross-domain transfer, annotator disagreement, and calibration rather than only aggregate accuracy.

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

7. Risks, Limitations, and Governance

7.1 Epistemic limitations

LSR is an integrative framework, not a new universally validated algorithm. Its scope is deliberately broad, grouping heterogeneous transformations under a shared separation-of-responsibilities principle. The framework becomes scientifically meaningful only when a study specifies the target relation, data-generating assumptions, loss, controls, and acceptance thresholds. The reviewed studies provide convergent architectural evidence but do not establish that every refinement mechanism will improve every domain.

7.2 Leakage and transductive contamination

Graph construction, clustering, normalization statistics, hard-negative mining, and taxonomy induction can leak information from validation or test data. Temporal applications are especially vulnerable when future edges or document versions enter training. Every preprocessing operation must state which split it observes. If a transductive setting is intended, it must be labeled and compared only with transductive baselines.

7.3 Bias and contestability

Rules and structure are not inherently neutral. Institutional categories can encode historical inequity; metadata may act as a proxy for protected attributes; graph message passing can propagate bias from connected nodes. Model cards should document intended use, excluded uses, affected groups, known disparities, data provenance, and appeal mechanisms. The ability to deactivate a refinement channel is valuable when a policy, taxonomy, or legal basis changes.

7.4 Versioning and operational compatibility

Encoder, refinement layer, taxonomy, normalization, and vector index form one deployable compatibility unit. Rolling back only the model while retaining vectors produced by a newer version creates silent incompatibility. A registry should enforce identifiers and dimensions at read and write time. Canary reindexing, dual reads, drift monitoring, and reversible migrations are safer than in-place replacement of an entire corpus.

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

7.5 Interpretability claims

A named cluster is not an interpreted latent dimension, and a stable nearest neighbor is not a causal explanation. Useful explanation artifacts include prototypes, counterfactual edits, relation paths, partial scores by view, uncertainty, and documented failure modes. Human evaluation should measure whether explanations support correct decisions, not merely whether they appear plausible.

7.6 Prompt sensitivity and semantic drift

Prompt-conditioned geometry can drift without changes to the underlying documents. Rewording an instruction, changing examples, altering prompt order, upgrading the encoder, or changing the number or placement of virtual prompt tokens may move vectors and invalidate an index. Soft prompts are additionally tied to embedding dimension, layer structure, tokenizer-independent positional conventions, and sometimes model-specific key–value interfaces. Robust deployment therefore requires prompt-level regression tests, neighborhood-change thresholds, dual-index migration when corpus vectors are affected, explicit token-budget monitoring, and rejection of incompatible prompt/model combinations.

8. Reproducibility and Acceptance Framework

- Freeze the test set before architecture and hyperparameter selection.
- Publish preprocessing, split identifiers, seeds, model and tokenizer revisions, package versions, and hardware.
- Cache base embeddings with cryptographic hashes and record normalization and pooling.
- Use parameter-matched controls and random-structure ablations.
- Report multiple seeds, uncertainty intervals, absolute and relative differences, and failed configurations.
- Measure end-to-end latency as well as encoder-only, refinement-only, indexing, and reindexing cost.
- Audit out-of-distribution data, subgroups, abstention behavior, and semantic counterexamples.
- Version encoder, refinement, taxonomy, and index as a single compatibility contract.
- For prompt-conditioned systems, publish the exact instructions or soft-prompt artifact hashes and their placement in the model.

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

- Evaluate semantic paraphrases, irrelevant-control prompts, prompt length, ordering, language, and unseen instructions.
- Report whether corpus documents were re-embedded for each prompt and verify query–document metric compatibility.
- Treat prompt, encoder, pooling, normalization, and index versions as one deployable contract.

Table 5. Minimum acceptance criteria for deploying a refinement layer.

Dimension	Minimum criterion	Required evidence
Effectiveness	Consistent improvement on a preregistered primary metric	Held-out predictions, multi-seed summary, confidence interval
Attribution	Benefit decreases under structure-removal/randomization controls	Parameter-matched and randomized-structure ablations
Preservation	No unacceptable loss on general or auxiliary capabilities	Transfer probes and neighborhood-preservation analysis
Robustness	No disproportionate degradation under plausible shift	Temporal/source splits, perturbations, and subgroup results
Efficiency	Quality gain satisfies a documented latency/memory SLO	End-to-end benchmark with hardware and batch details
Governance	Every output is traceable to compatible versions and data	Registry, model card, provenance logs, rollback test
Interpretability	Explanations are stable and useful to intended reviewers	Human protocol, stability analysis, counterexamples

9. Discussion

The principal value of layered refinement is architectural clarity and an explicit evaluation contract. It avoids asking a single encoder to be simultaneously universal, domain-specific,

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

hierarchical, interpretable, compressed, and optimized for every retrieval relation. Instead, it treats general semantics as a substrate and makes downstream assumptions explicit. This design resembles modular statistical modeling: each component has a defined role, can be ablated, and can fail independently. The framework should be judged as an integration pattern and protocol unless a concrete LSR instantiation demonstrates gains over the strongest multi-view, instruction-conditioned, multi-vector, and structure-aware baselines.

The framework also changes how improvement should be interpreted. If a nonlinear refinement layer followed by a classifier outperforms logistic regression on h, the result may reflect extra classifier capacity rather than a better reusable embedding. Genuine representational improvement requires additional evidence: stronger linear probes on z, better metric retrieval, stable neighborhoods for the target relation, preservation of non-target information, and benefits that persist when downstream capacity is controlled. Conversely, a refinement can be operationally valuable even if it deliberately sacrifices generality; the claim must simply be scoped to that task and documented.

Prompt embeddings extend this argument by making the geometry conditional. The phrase “the embedding of x” becomes incomplete when the representation also depends on instruction I, prompt matrix P, model version, tokenizer, and pooling rule. The scientifically relevant object is the conditional representation h(x|I,P,model version), together with the relation it is intended to encode. A prompt token is an input position; a prompt embedding is its continuous vector or the pooled representation of the instruction; and the final text embedding is the representation used by the consumer. This distinction turns prompt design from an informal interface choice into part of the measurement model. A short prompt token bank can reduce repeated textual context, but it does not eliminate the computational cost of its virtual positions and does not automatically reduce generation length.

Future research should examine query-conditioned mixtures of refinement views, continual refinement under taxonomy drift, uncertainty-aware routing, hyperbolic or product-manifold representations for non-Euclidean structures, and causal tests that distinguish semantic information from shortcuts. Matryoshka-style dimensional nesting is particularly promising for trading retrieval quality against memory without maintaining independent models. Another priority is benchmark design: datasets should expose multiple relations over the same items so that a system can demonstrate when one geometry is insufficient.

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

10. Conclusion

Organized embeddings should be understood as engineered representation systems rather than vectors that happen to form visually appealing clusters. A general encoder provides broad semantic competence; refinement layers expose or inject task-relevant relations; external structures make hierarchy and connectivity queryable; prompt tokens or instructions condition which relation is emphasized; and explicit losses define what the system is trying to preserve or change. The operational pipeline is token sequence → token embeddings → contextual token states → pooled base embedding → relation-specific refinement → indexed or generated consumer. TabEmb, TAROT, TopicForest, hierarchical multitask learning, Matryoshka-based clustering, multi-view retrieval, and late-interaction retrieval demonstrate complementary instances of this principle.

The layered semantic refinement framework offers a practical path toward task-appropriate, versioned, and auditable vector representations. Its scientific standard is deliberately demanding: the intended relation must be defined, measured in the original space, tested under shift, compared with capacity-matched controls, and attributed through ablation. Until the proposed AG News protocol or a domain-specific benchmark is executed, this article should be read as a technical-methodological synthesis and research blueprint, not as an empirical report of performance gains.

Declarations

Funding: No funding information was provided for this technical-methodological article.

Conflict of interest: No conflict-of-interest information was provided.

Data and code availability: No new dataset was generated and no benchmark was executed. The implementation excerpts define a reproducible protocol to be completed in a future empirical study.

Ethics statement: The article analyzes computational methods and does not report research involving human participants or animals.

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

References

Zhang, S., Liang, Y., Gong, M., Jiang, D., & Duan, N. (2022). Multi-view document representation learning for open-domain dense retrieval. In Proceedings of ACL 2022, 5990–6000. https://doi.org/10.18653/v1/2022.acl-long.414

Santhanam, K., Khattab, O., Saad-Falcon, J., Potts, C., & Zaharia, M. (2022). ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In Proceedings of NAACL-HLT 2022, 3715–3734. https://doi.org/10.18653/v1/2022.naacl-main.272

Ruan, Q., Ostendorff, M., & Rehm, G. (2022). HiStruct+: Improving extractive text summarization with hierarchical structure information. Findings of ACL 2022, 1292–1308. https://doi.org/10.18653/v1/2022.findings-acl.102

Kabakum, I., Montgomery, T., Ravenwood, D., & Harrington, G. (2025). Semantic layered embedding diffusion in large language models for multi-contextual consistency. arXiv:2501.15405. https://arxiv.org/html/2501.15405v1

Cao, Y., Chen, X., Du, L., Chen, H., Fu, Q., Han, S., Du, Y., Kang, Y., Lu, G., & Li, Z. (2024). TAROT: A hierarchical framework with multitask co-pretraining on semi-structured data towards effective person-job fit. In 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 12046–12050). IEEE. https://doi.org/10.1109/ICASSP48485.2024.10447647

Chang, C.-H., Ondov, B., Choi, B., Peng, X., He, H., & Xu, H. (2025). TopicForest: Embedding-driven hierarchical clustering and labeling for biomedical literature. Journal of Biomedical Informatics, 172, 104958. https://doi.org/10.1016/j.jbi.2025.104958

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of NAACL-HLT 2019 (pp. 4171–4186). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

Gao, T., Yao, X., & Chen, D. (2021). SimCSE: Simple contrastive learning of sentence embeddings. In Proceedings of EMNLP 2021 (pp. 6894–6910). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.552

Hanley, H. W. A., & Durumeric, Z. (2025). Hierarchical level-wise news article clustering via multilingual Matryoshka embeddings. In Proceedings of the 63rd Annual Meeting of the

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

Association for Computational Linguistics (Vol. 1, pp. 2476–2492). https://doi.org/10.18653/v1/2025.acl-long.124

Hoseinzade, E., Wang, K., & Raju, A. D. (2026). TabEmb: Joint semantic-structure embedding for table annotation. In Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Vol. 1, pp. 16620–16631). https://doi.org/10.18653/v1/2026.acl-long.757

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In International Conference on Learning Representations. https://arxiv.org/abs/1609.02907

Kusupati, A., Bhatt, G., Rege, A., Wallingford, M., Sinha, A., Ramanujan, V., Howard-Snyder, W., Chen, K., Kakade, S., Jain, P., & Farhadi, A. (2022). Matryoshka representation learning. In Advances in Neural Information Processing Systems, 35, 30233–30249.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459–9474.

Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. In Proceedings of EMNLP 2021 (pp. 3045–3059). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.243

Li, X. L., & Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. In Proceedings of ACL-IJCNLP 2021 (Vol. 1, pp. 4582–4597). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.353

Liu, X., Ji, K., Fu, Y., Tam, W. L., Du, Z., Yang, Z., & Tang, J. (2022). P-Tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. In Proceedings of ACL 2022 (Short Papers) (pp. 61–68). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-short.8

Maaten, L. van der, & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9, 2579–2605.

CC BY

Nexus Science Review
INTERNATIONAL JOURNAL OF INTERDISCIPLINARY RESEARCH

Received: 10/06/2026 - **Accepted:** 17/08/2026
Vol: 01.03 - August
DOI: 10.66104/ttk33814
Pages: 1-34

McInnes, L., Healy, J., & Melville, J. (2018). UMAP: Uniform manifold approximation and projection for dimension reduction. arXiv:1802.03426. https://doi.org/10.48550/arXiv.1802.03426

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Proceedings of EMNLP-IJCNLP 2019 (pp. 3982–3992). Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1410

Sanh, V., Wolf, T., & Ruder, S. (2019). A hierarchical multi-task approach for learning embeddings from semantic tasks. Proceedings of the AAAI Conference on Artificial Intelligence, 33(1), 6949–6956. https://doi.org/10.1609/aaai.v33i01.33016949

Su, H., Shi, W., Kasai, J., Wang, Y., Hu, Y., Ostendorf, M., Yih, W.-T., Smith, N. A., Zettlemoyer, L., & Yu, T. (2023). One embedder, any task: Instruction-finetuned text embeddings. In Findings of the Association for Computational Linguistics: ACL 2023 (pp. 1102–1121). https://doi.org/10.18653/v1/2023.findings-acl.71

Peng, L., Zhang, Y., Wang, Z., Srinivasa, J., Liu, G., Wang, Z., & Shang, J. (2024). Answer is all you need: Instruction-following text embedding via answering the question. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Vol. 1, pp. 459–477). https://doi.org/10.18653/v1/2024.acl-long.27

Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., & Wei, F. (2024). Improving text embeddings with large language models. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Vol. 1, pp. 11897–11916). https://doi.org/10.18653/v1/2024.acl-long.642

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In Advances in Neural Information Processing Systems, 28, 649–657.

CC BY