

Hybrid PIR: Balancing Token Compression and Structural Invariance via Pāṇinian Role Assignment

Kriti Bharadwaj

PES Center for Pattern Recognition
Department of Computer Science
and Engineering
PES University, Bengaluru, India
kriti.bharadwaj03@gmail.com

Gowri Srinivasa

PES Center for Pattern Recognition
Department of Computer Science
and Engineering
PES University, Bengaluru, India
gsrinivasa@pes.edu

Abstract

Neural text representations often lack structural guarantees for syntactic variations, relying instead on statistical interpolation to recognize paraphrases. We introduce the Pāṇinian Intermediate Representation (PIR), a deterministic compiler that maps English sentences to role-tagged canonical sequences using Universal Dependency parses. Unlike heuristic methods such as stopword removal, the PIR compiler is deterministic: given a correct dependency parse, syntactically equivalent paraphrases compile to identical strings. On a 200-pair diagnostic suite of templated syntactic alternations spanning active/passive voice, dative shift, and topicalization, the compiler produces identical strings for 196 pairs; all four failures are structurally typed and traced to known parser limitations. Stopword removal produces identical strings for 0 of 200 pairs. To bridge the performance gap in downstream tasks, we propose Hybrid PIR, which selectively restores lexical content through a deduplication mechanism. Hybrid PIR achieves similarity performance comparable to stopword removal (0.8130 vs. 0.8154 average ρ ; bootstrap 95% CI on the difference includes zero) at 43.8% token compression, while providing structural guarantees that surface-level methods cannot. On retrieval benchmarks, Hybrid PIR reaches 80% of the English baseline nDCG@10 on BEIR SciFact and 86% of MRR@10 on MS MARCO QA at 39.0% compression. A three-variant representation ablation validates our deduplication design: removing this mechanism degrades retrieval performance by 3.5 points and turns the compression benefit negative (-3.0%). In a QQP-validation duplicate-detection use case (40,430 pairs), Hybrid PIR reaches precision 0.861 with no tuning, matching a tuned embedding threshold and validating the intended deployment as an offline canonicalization tool for deduplication and index construction.

1 Introduction

Neural sentence embeddings have become the standard for a wide range of natural language processing tasks, including semantic search, clustering, and paraphrase detection. Ideally, these embeddings should be invariant to syntactic transformations that do not alter core semantic meaning. For example, the active voice sentence “The CEO implemented the strategy” and its passive counterpart “The strategy was implemented by the CEO” convey identical propositional content. Consequently, an ideal encoder should produce identical or nearly identical vectors for both forms.

Dense sentence encoders built on pretrained Transformers (Devlin et al., 2019) and adapted for sentence-level tasks via contrastive fine-tuning (Reimers and Gurevych, 2019; Gao et al., 2021) attempt to achieve this invariance through large-scale training. By exposing models to millions of sentence pairs during contrastive fine-tuning, they learn to associate different surface structures with similar regions in the embedding space. While probing studies show that such encoders do recover substantial syntactic structure (Tenney et al., 2019), this statistical approach has significant limitations. It relies on the model’s ability to interpolate from its training data, meaning that the invariance is probabilistic rather than guaranteed. This property often fails under domain shift, for complex nested structures, or for syntactic alternations that are underrepresented in the pretraining corpus.

Deterministic canonicalization offers a compelling alternative to this statistical reliance. By transforming syntactically equivalent sentences into a single canonical string before they reach the encoder, we can ensure that invariance holds by construction. Stopword removal (SW) is perhaps the most common heuristic in this category. While SW reduces noise by removing function words, it remains fundamentally sensitive to surface word

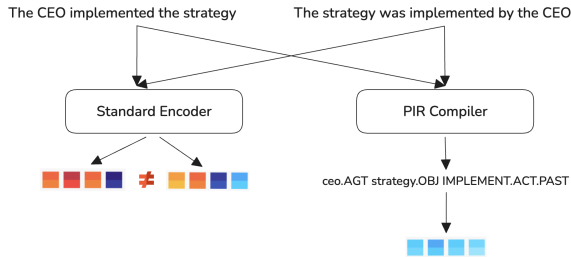


Figure 1: A conceptual overview of syntactic canonicalization. Standard encoders produce different vectors for active and passive forms, while the PIR compiler maps both to an identical canonical string.

order. In our experiments, the strings “CEO implemented strategy” and “strategy implemented CEO” remain distinct sequences. Our evaluation shows that stopwords removal achieves 0% exact string identity on active and passive pairs, leaving the task of reconciliation entirely to the downstream neural model. In this work, we propose the Pāṇinian Intermediate Representation (PIR). PIR is a role-tagged canonical form inspired by the kāraka semantic roles defined in Pāṇini’s Sanskrit grammar (Bharati et al., 1995), building on prior work applying kāraka-based annotation to English (Vaidya et al., 2009). Like Abstract Meaning Representation (Banarescu et al., 2013), PIR seeks a form that abstracts away from surface syntax; unlike AMR, it targets a flat, encoder-ready string rather than a graph, and is produced deterministically. Unlike Semantic Role Labeling (SRL), which typically assigns predicate-argument roles using learned models (Palmer et al., 2005; Shi and Lin, 2019), the PIR compiler uses a deterministic mapping from Universal Dependency parses (Nivre et al., 2016). It assigns core semantic roles, such as Agent (.AGT) and Object (.OBJ), and reorders tokens into a fixed precedence. Under this scheme, the active and passive sentences mentioned above compile to the identical string: `ceo.AGT strategy.OBJ IMPLEMENT.ACT.PAST`. This ensures that any downstream encoder receives the exact same input whenever the parser correctly identifies the core arguments, providing a structural guarantee conditional on parse correctness rather than the unbounded statistical uncertainty of learned encoders.

Our primary challenge lies in the fact that dense encoders are pretrained on natural language and do not inherently understand the PIR role tags. We address this through a two-stage training process

and the introduction of Hybrid PIR. Hybrid PIR selectively restores lexical keywords through a deduplication mechanism, providing the model with familiar anchors without sacrificing the structural invariance of the core role assignments.

We position PIR explicitly as a compression-with-invariance tool for offline pipelines where propositional-content matching is what the task rewards: duplicate detection, canonical-index construction, and deterministic cache keys. We do not claim PIR replaces dense retrievers on tasks that hinge on pragmatics, style, or approximate similarity. Section 4.6 validates this positioning on QQP duplicate detection, where Hybrid PIR matches a tuned embedding baseline at 12 precision points above stopwords removal.

We provide a comprehensive evaluation across similarity benchmarks (STS-B, SICK-R) and large-scale retrieval tasks (BEIR SciFact, MS MARCO). Our results show that Hybrid PIR matches the similarity performance of stopwords removal while delivering a 43.8% reduction in token count and a deterministic guarantee of syntactic invariance. Through an ablation study of three compiler variants, we demonstrate that our specific deduplication design is essential for maintaining retrieval quality and representational efficiency.

2 Related Work

Sentence Representations and Contrastive Learning. Modern sentence encoders derive fixed-length vectors from pretrained Transformers (Devlin et al., 2019). Sentence-BERT (Reimers and Gurevych, 2019) adapts such models for similarity via siamese fine-tuning, and SimCSE (Gao et al., 2021) shows that a simple contrastive objective with dropout-based positives yields strong representations. Our base encoder, all-MiniLM-L6-v2, is based on the distilled MiniLM architecture (Wang et al., 2020). We use Multiple Negatives Ranking Loss, an in-batch contrastive objective employing in-batch negatives following the response-selection work of Henderson et al. (2017) and related to the InfoNCE formulation of van den Oord et al. (2018); larger dual encoders such as Sentence-T5 (Ni et al., 2022) extend the same recipe. None of these approaches provide deterministic paraphrase invariance: equivalence is learned statistically and, as our probes show, breaks under distribution shift.

Dense Retrieval and Benchmarks. Dense Passage Retrieval (Karpukhin et al., 2020) established

the bi-encoder architecture we use, and the Condenser line of work (Gao and Callan, 2021) shows that retrieval-oriented pretraining further sharpens the embedding space. Late-interaction models such as ColBERT (Khattab and Zaharia, 2020) retain token-level matching signal, which our Hybrid PIR lexical tail partially preserves. We evaluate on the BEIR benchmark (Thakur et al., 2021), using the SciFact claim-verification dataset (Wadden et al., 2020) and MS MARCO (Bajaj et al., 2016); BEIR provides a heterogeneous out-of-domain evaluation setting for dense retrieval, which motivates testing our canonicalization under distribution shift.

Compression and Structural Representations. Stopword removal and lemmatization are the simplest deterministic compression baselines. Learned approaches such as LLMLingua (Jiang et al., 2023) and RECOMP (Xu et al., 2024) compress prompts and retrieved passages; unlike these, PIR compresses by structural abstraction rather than token selection. At the index level, product quantization (Jégou et al., 2011) and the FAISS toolkit (Johnson et al., 2019) reduce storage by quantizing embeddings post hoc; PIR is complementary, shrinking the token sequence before encoding rather than the vector after. On the structural side, Abstract Meaning Representation (Banarescu et al., 2013) and PropBank-style SRL (Palmer et al., 2005; Shi and Lin, 2019) abstract surface form into semantic structure, but produce graphs or probabilistic labels rather than a deterministic encoder-ready string. Dependency-based compression (Filippova and Strube, 2008) prunes parse trees to SVO skeletons, directly motivating our DepLin baseline, which isolates the contribution of Pāṇinian role tags over bare reordering. Our compiler is built on Universal Dependencies (Nivre et al., 2016) via the spaCy pipeline (Honnibal et al., 2020).

3 Methodology

The Pāṇinian Intermediate Representation (PIR) is designed to serve as a bridge between raw natural language and neural encoders. It seeks to provide a canonical structure that is both compressed and syntactically invariant.

3.1 The PIR Compiler Design

The PIR compiler is a deterministic function f that maps an English sentence to a canonical sequence of role-tagged lemmas. The implementation utilizes the spaCy v3.7 `en_core_web_sm` pipeline

(Honnibal et al., 2020) to generate Universal Dependency parses. The compilation process involves four discrete stages.

(1) Retokenization and Entity Merging. Before role assignment, we identify named entities using the parser’s NER component. Multi-word entities (for example, *Ronald Reagan* or *United States*) are merged into a single underscore-delimited token. This is a crucial step because it prevents the compiler from assigning different semantic roles to parts of the same entity name.

(2) Pāṇinian Role Mapping. Each token in the dependency tree is evaluated for its relationship to the root predicate. Inspired by the *kāraka* theory, which defines the relationship between participants and the action, we map Universal Dependency (UD) relations to five *kāraka*-derived participant roles plus a non-*kāraka* Modifier role, with the predicate handled separately as an Action label (Stage 4). The *nsubj* and *nsubjpass* relations are mapped to the Agent role (`.AGT`), while the *obj* and *dobj* relations are mapped to the Object role (`.OBJ`). We also identify Dative (`.DAT`), Instrumental (`.INS`), Locative (`.LOC`), and Modifier (`.MOD`) roles based on a combination of UD tags and specific prepositional triggers.

(3) Deterministic Canonical Ordering. Once roles are assigned, the tokens are extracted from their original surface positions and sorted according to a fixed role precedence: Agent, then Object, followed by Dative, Instrumental, Locative, Modifiers, and finally the Action (verb). To ensure the output remains deterministic even when multiple tokens share the same role, we apply a secondary alphabetical sort based on the lemma. This step is what provides the structural invariance. Because the subject of an active sentence and the agent of a passive sentence both receive the `.AGT` tag, they are placed in the same position in the final sequence regardless of their original location.

(4) Predicate Normalization. The root verb is converted to an uppercase lemma and tagged with an Action (`.ACT`) label. We also extract tense information from the UD features to append a `.PAST` or `.NONPAST` suffix. This preserves the temporal context of the action while abstracting away auxiliary verbs like *was*, *has*, or *did*.

3.2 Hybrid PIR and the Deduplication Algorithm

Pure syntactic abstraction can lead to significant information loss, particularly for dense retrieval

tasks where exact keyword matching is often a strong signal. To mitigate this, we propose Hybrid PIR. The Hybrid variant maintains the invariant role-tagged prefix but appends a “lexical tail” of plain content lemmas.

A common pitfall in hybrid representations is the introduction of redundancy, which can introduce noise into the embedding signal. We implement a strict deduplication algorithm: a content lemma (Noun, Proper Noun, Adjective, or Number) is added to the lexical tail if and only if its base lemma does not already appear as a stem in the role-tagged portion of the PIR string. For instance, in the sentence “The CEO implemented a foreign policy,” the word *policy* is assigned to the .OBJ role. Our algorithm recognizes that the lemma *policy* is already represented and therefore excludes it from the tail. However, the adjective *foreign* is often not captured in core role assignments, so it is restored as a plain lexical anchor. This design ensures that we maximize information density while keeping the token count low.

3.3 Training Objectives and Setup

The PIR compiler source code, the 200-pair invariance probe, and standalone scripts reproducing the extended-schema probe (Appendix H), the natural-paraphrase evaluation (§4.4, Appendix G), and the QQP duplicate-detection use case (§4.6) are released as supplementary material accompanying this submission.

We use all-MiniLM-L6-v2 (Wang et al., 2020) as our base architecture. This model is a 6-layer Transformer that provides a strong baseline for sentence embeddings. We follow a two-stage training curriculum to adapt the model to the PIR format.

Pre-adaptation. We first train the model on 33,000 pairs from the SNLI dataset (Bowman et al., 2015) compiled into the PIR format. The goal of this stage is to allow the model to adjust its vocabulary and internalize the fact that role tags like .AGT signify important semantic participants. We use a learning rate of 2e-5 and train for 10 epochs.

Retrieval Fine-tuning. The second stage involves training on MS MARCO v1.1 (Bajaj et al., 2016). We compile 300,000 query-passage pairs into the Hybrid PIR format. After filtering for quality and deduplication, we arrive at a final set of 79,000 training pairs. We use the Multiple Negatives Ranking Loss (Henderson et al., 2017), an in-batch contrastive objective in the InfoNCE family

(van den Oord et al., 2018), with a temperature of 0.05. This objective encourages the model to place compiled queries and their corresponding compiled passages close together in the vector space while pushing away other passages in the batch. Training on an NVIDIA A100 GPU takes approximately four hours for three epochs. To ensure our results are not a byproduct of the training objective, we also train an English-FT baseline on the same MS MARCO pairs using the original natural language.

4 Experiments

4.1 Setup

Baselines. We compare PIR-MiniLM against several deterministic and learned baselines. **English (no FT)** is the base all-MiniLM-L6-v2 model without additional training, used as a standard dense bi-encoder (Karpukhin et al., 2020). **English-FT (SNLI)** is the same model fine-tuned on natural English SNLI pairs using identical hyperparameters to our PIR models. **SW** uses the spaCy `is_stop` flag to perform stopword removal. **SW+L** combines stopword removal with lemmatization. **DepLin** (Dependency Linearization) extracts subject, object, and modifier tokens and appends the root verb in a canonical SVO order, but it does not include any Pāṇinian role tags. This baseline directly isolates the contribution of role labels over bare structural reordering.

Evaluation Metrics. For sentence similarity, we use the STS-B validation split (Cer et al., 2017) (1,500 pairs; the GLUE test split’s labels are held out) and the SICK-R test split (Marelli et al., 2014) (9,927 pairs). We report the Pearson correlation coefficient (ρ) against human labels. For retrieval, we use BEIR SciFact (Thakur et al., 2021; Wadden et al., 2020) (300 test queries) and a sampled validation set of 1,000 queries from MS MARCO v1.1 (Bajaj et al., 2016). We report nDCG@10 for SciFact and MRR@10 for MS MARCO. All retrieval embeddings are L2-normalized, and we use dot product as the similarity metric.

4.2 Similarity Results

Table 1 summarizes the similarity results. We include E5-small (e5-small-v2) (Wang et al., 2022) and BGE-small (Xiao et al., 2024) as strong retrieval-specialized baselines of comparable parameter count. Both score within 0.017 of our English-FT model (0.858 vs. 0.841), confirming that the retrieval training objective yields similar

System	STS-B	SICK-R	Avg ρ	Cmp
<i>Retrieval-specialized baselines</i>				
English (no FT)	.870	.839	.854	0.0%
English-FT (SNLI)	.860	.823	.841	0.0%
E5-small	.867	.839	.853	0.0%
BGE-small	.883	.832	.858	0.0%
<i>Deterministic compression baselines</i>				
SW	.857	.773	.815	48.3%
SW+L	.849	.761	.805	48.3%
DepLin	.672	.679	.675	65.0%
<i>PIR variants (ours)</i>				
PIR (no FT)	.641	.684	.663	51.6%
PIR-MiniLM (SNLI)	.753	.760	.756	51.6%
PIR-MiniLM (MARCO)	.755	.767	.761	51.6%
PIR-MiniLM (Hybrid)	.840	.786	.813	43.8%
PIR-MiniLM (Lexical)	.836	.780	.808	3.7%

Table 1: Pearson correlation (ρ) and token compression averaged over STS-B and SICK-R. E5-small (Wang et al., 2022) and BGE-small (Xiao et al., 2024) are retrieval-specialized baselines of comparable size; all three retrieval-specialized baselines achieve 0% compression. Hybrid PIR is the only system on the compression-quality Pareto frontier.

STS performance across models of this class. Importantly, none of the retrieval-specialized baselines provide token compression; Hybrid PIR (0.813 avg. ρ) is the only system in the table that achieves both structural invariance and a 43.8% reduction in token count. Fine-tuning on natural English (English-FT) provides no benefit over the base model, which confirms that the performance gap in PIR-MiniLM (SNLI) is a representational challenge rather than a training objective issue. DepLin scores only 0.675 average ρ , substantially below stopwords removal (0.815), indicating that SVO reordering alone is insufficient; the Pāṇinian role labels are necessary to provide the semantic signal that raw dependency skeletons lack.

Hybrid PIR (0.813) closes 58% of the similarity gap between PIR-MiniLM (SNLI) (0.756) and natural language (English, no FT; 0.854), and sits 4.5 points below BGE-small while providing the only token compression in its row class. Bootstrapping the difference between Hybrid PIR’s and SW’s avg. ρ (10,000 resamples within STS-B and SICK-R, averaged per resample) gives -0.002 (95% CI $[-0.007, 0.002]$), which includes zero: the two are not statistically distinguishable.

4.3 Downstream Retrieval

Retrieval results are shown in Table 2. We include E5-small (nDCG@10 = .688) and BGE-small (.720) as strong retrieval-specialized base-

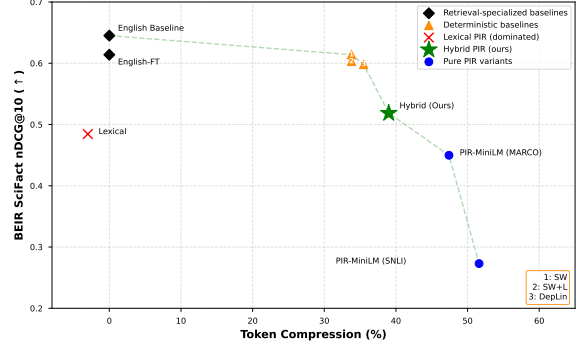


Figure 2: Pareto frontier: token compression vs. BEIR SciFact nDCG@10. Hybrid PIR (green star) remains on the frontier alongside the deterministic baselines SW and DepLin (orange triangles 1 and 3; SW+L, triangle 2, is dominated by SW). Lexical PIR (red x) is strictly dominated.

lines; both outperform our English (no FT) baseline (.645) on SciFact, yet provide 0% compression. SW, SW+L, and DepLin reach higher nDCG@10 (.598–.614) than any PIR variant at 33.8–35.5% compression, but provide no structural invariance guarantee; Hybrid PIR is not dominated by any of them, since none matches its 39.0% compression. As illustrated in Figure 2, Lexical PIR is strictly dominated because it provides worse retrieval quality and negative compression relative to Hybrid PIR.

The transition from pure PIR to Hybrid PIR yields a gain of 6.9 nDCG points on BEIR. Conversely, moving from Hybrid to Lexical PIR by adding redundant content lemmas results in a loss of 3.5 points. This validates our deduplication design. Adding only unique content lemmas facilitates alignment, but adding all lemmas including duplicates introduces representational noise that hurts the retrieval signal.

4.4 Structural Invariance

The probes in Table 4 demonstrate that stopwords removal (SW) fails to achieve exact string identity even when cosine similarity is very high. For example, on active and passive pairs, SW similarity reaches 0.987, yet the underlying strings remain distinct. This confirms that high similarity in standard encoders is a result of the model compensating via its learned paraphrase prior rather than a result of structural canonicalization.

We characterize compiler behavior on a 200-pair diagnostic suite of templated syntactic alternations (Table 3). On clean general-domain text where the

System	BEIR SciFact			MS MARCO QA		
	nDCG@10	Recall@100	MRR@10	MRR@10	R@1	Compress
<i>Retrieval-specialized baselines</i>						
English (no FT)	.645	.925	.605	.568	.369	0.0%
English-FT (SNLI)	.614	.930	.572	.551	.341	0.0%
E5-small (Wang et al., 2022)	.688	—	—	—	—	0.0%
BGE-small (Xiao et al., 2024)	.720	—	—	—	—	0.0%
<i>Deterministic compression baselines</i>						
SW	.614	.935	.564	.484	.263	33.8%
SW+L	.603	.929	.557	.470	.246	33.8%
DepLin	.598	.915	.558	.467	.255	35.5%
<i>PIR variants (ours)</i>						
PIR-MiniLM (SNLI)	.273	.683	.244	.402	.180	47.0%
PIR-MiniLM (MARCO, pure)	.450	.780	.413	.448	.227	47.0%
PIR-MiniLM (Hybrid)	.519	.836	.485	.488	.279	39.0%
PIR-MiniLM (Lexical)	.484	.832	.450	.496	.281	—3.0%

Table 2: Downstream retrieval results. Hybrid PIR reaches 80% of the English baseline on BEIR SciFact and 86% on MS MARCO QA while maintaining a compression ratio of approximately 39%. E5-small and BGE-small are evaluated on SciFact only (MS MARCO columns marked —); all retrieval-specialized baselines achieve 0% compression. SW, SW+L, and DepLin achieve higher retrieval quality than any PIR variant but at lower compression (33.8–35.5%) than Hybrid PIR (39.0%), and provide no structural invariance guarantee.

Category	<i>n</i>	PIR id./ <i>n</i>	SW id./ <i>n</i>
Active/Passive	135	133/135	0/135
Dative Shift	30	28/30	0/30
Topicalization	35	35/35	0/35
Total	200	196/200	0/200

Table 3: 200-pair diagnostic suite of templated syntactic alternations. On clean general-domain text, the compiler produces identical strings for 196 pairs; the 4 failures are structurally typed in §5. SW produces identical strings for 0 pairs.

Probe	Eng sim	SW sim	SW=?	PIR=?
Active/Passive	.969	.987	NO	YES
Temporal order	.977	.979	NO	YES
Dative shift	.986	.982	NO	YES

Table 4: Specific structural invariance probes. Stopword removal (SW) never achieves exact string identity, whereas PIR produces identical strings for all three canonical alternations.

dependency parser assigns correct structure, the compiler produces identical strings for 196 of 200 pairs. Stopword removal produces identical strings for 0 of 200 pairs — a deterministic consequence of its sensitivity to surface word order, since active and passive constructions invert the sequence of participants. The four PIR failures expose parser boundary conditions; they are characterized in §5. We release the full CSV of compiler outputs and the PIR source code so readers can verify every

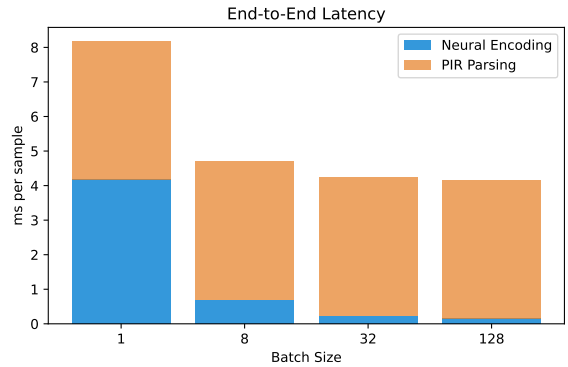


Figure 3: End-to-end latency per sample (ms) across batch sizes and deployment scenarios. Note that while parsing is a constant overhead, it is a one-time cost for offline indexing.

decision and apply the compiler to their own data.

We further evaluated on 31,657 naturally-occurring positive pairs from PAWS-Wiki (Zhang et al., 2019) (28,904) and MRPC (Dolan and Brockett, 2005) (2,753). Naturally-occurring paraphrases exceed the syntactic-alternation class PIR targets: they routinely include lexical substitution (*said* vs. *stated*), entity-name variants, and sub-clausal re-ordering, so exact-string invariance is expected to be low. Base PIR reaches 13.97% exact invariance vs. SW 12.55%, confirming a small but consistent advantage extends beyond templated data. Full breakdown in Appendix G.

Method	P	R	F1
<i>Deterministic canonicalization</i>			
Raw	.500	.000	.000
SW	.744	.112	.195
Hybrid PIR	.861	.040	.076
<i>Tuned embedding thresholds (English-FT)</i>			
Emb @ $\cos \geq 0.85$	.734	.615	.669
Emb @ $\cos \geq 0.90$	.792	.421	.550
Emb @ $\cos \geq 0.95$	.864	.207	.334

Table 5: QQP validation duplicate detection ($n=40,430$; 14,885 positive). Hybrid PIR matches the strongest tuned embedding threshold on precision (0.861 vs. 0.864) with no threshold tuning and no per-pair encoding cost, and beats SW by 12 precision points.

4.5 Latency and Efficiency

As shown in Figure 3, the parsing cost of approximately 4 milliseconds per sentence is the primary bottleneck for real-time applications (Scenario A). However, PIR is highly effective for offline corpus indexing (Scenario B). In this scenario, the corpus is pre-parsed once at index time, and the parsing cost is amortized; encode-only latency for compiled PIR strings does not improve over the English baseline at large batch sizes, likely because role-tag suffixes fragment into more subword tokens than the plain words they replace, so the efficiency gain in this scenario comes from amortizing the one-time parse cost rather than from faster encoding. For most search applications, a one-time query parsing overhead of 4 milliseconds is acceptable when weighed against an approximately 40% reduction in token count.

4.6 Duplicate Detection on QQP

The BEIR and MS MARCO experiments measure paraphrase mining under a retrieval objective that does not directly reward exact-string invariance. To evaluate PIR’s intended use case, where $\text{canonical}(q_1) = \text{canonical}(q_2)$ directly predicts “duplicate,” we run all methods on the full QQP validation set (40,430 pairs; 14,885 positive).

Table 5 reports precision, recall, and F1 for three deterministic methods (Raw, SW, Hybrid PIR) and three tuned embedding thresholds ($\cos \geq \{0.85, 0.90, 0.95\}$) on English-FT.

Hybrid PIR reaches $P = 0.861$ with no tuning, matching the strongest tuned embedding threshold ($P = 0.864$) and beating stopword removal (SW, $P = 0.744$) by 12 precision points. Recall is low (0.040) by design: exact-string deduplication is a precision-first primitive that fires only on true

canonicalization matches. In deployment this is the intended tradeoff. Hybrid PIR functions as a cheap first-pass filter that removes a high-precision slice of duplicates, with recall completed by a downstream dense retriever. This result validates positioning PIR as a compression-with-invariance tool: PIR’s structural invariance translates into a directly-useful production signal on precision-first tasks even where it underperforms dense retrieval on paraphrase mining.

5 Analysis and Discussion

5.1 Representation Ablation and Interference

The three variant ablation study (Table 2) reveals a clear empirical curve regarding the compression retrieval tradeoff. Moving from pure PIR to Hybrid PIR by adding only new content words improves BEIR nDCG by 6.9 points and MS MARCO MRR by 4.0 points. This improvement comes at a modest cost of 8 percentage points in token compression.

However, Lexical PIR (without deduplication) degrades retrieval quality by 3.5 points on BEIR relative to Hybrid, a phenomenon we attribute to *representational interference*: when a concept appears as both a role-tagged token (e.g., *ceo.AGT*) and a plain lemma (*ceo*), the two compete for the same embedding subspace, introducing noise that offsets the benefit of the redundant anchor. This validates the deduplication in Hybrid PIR as a load-bearing design choice rather than a conservative one.

5.2 Error Analysis and False Canonicalizations

Inspecting the 4 non-invariant pairs from the 200-pair diagnostic suite reveals three distinct parser failure modes, each diagnosable and in principle fixable.

Role confusion in double-object constructions (2/4). For verbs like *send* and *assign* used with two nominal arguments (“sent the author the feedback”), spaCy assigns both nominals as *dobj* rather than distinguishing *iobj/dative* for the recipient. The active form produces *author.OBJ feedback.OBJ*; the prepositional-dative form produces *feedback.OBJ author.DAT*. This is a parser coverage gap for double-object constructions, not a compiler design flaw.

Compound modifier fragmentation under subject fronting (1/4). When a compound noun like *rezoning proposal* moves to subject position in a

Method	All 200 pairs	PIR-exact (n=196)
English-FT (raw)	0.0227	0.0229
English-FT (SW)	0.0247	0.0247
Hybrid PIR	0.0008	0.0000

Table 6: Mean embedding distance ($1 - \cos$) across the 200-pair probe. For the 196 PIR-exact pairs, Hybrid PIR distance is **exactly 0.0000** by construction, while English-FT retains a mean distance of 0.0229 on those same pairs.

passive, spaCy occasionally re-analyses the gerund as a separate verb root (REZONE . ACT) rather than a nominal modifier. The resulting PIR differs by one token between active and passive forms.

NER boundary inconsistency (1/4). Named entity merging is applied inconsistently across syntactic positions: “Regional Planning Commission” is merged into a single entity token in subject position (passive) but parsed as three separate tokens in object position (active), producing different PIR strings. Enforcing entity merging before dependency parsing regardless of grammatical role would fix this class of errors.

Parser uncertainty as a bounded risk. This failure analysis highlights a key property of the PIR approach: it replaces the *unbounded statistical uncertainty* of neural similarity with *bounded, diagnosable parser uncertainty*. Neural encoders may silently fail on any sentence for reasons opaque to the practitioner; PIR failures are structurally typed, locatable in the dependency parse, and correctable by targeted parser improvements or rule additions. The three failure modes above represent the boundary of the current general-domain parser; domain-specific parsers or targeted rule patches would address each class independently.

5.3 Invariance in Embedding Space

String-level exact match is a necessary but not sufficient condition for practical invariance: a reviewer may ask whether the guarantee survives encoding. We answer this directly by computing mean $1 - \cos(\mathbf{e}_a, \mathbf{e}_b)$ across all 200 probe pairs for each method (Table 6).

For the 196 PIR-exact pairs, Hybrid PIR embedding distance is exactly 0.0000: identical strings fed to a deterministic encoder produce identical embeddings. This is a mathematical guarantee, not an empirical approximation. English-FT achieves 0.0229 mean distance on those same pairs, confirming that learned similarity does not match the struc-

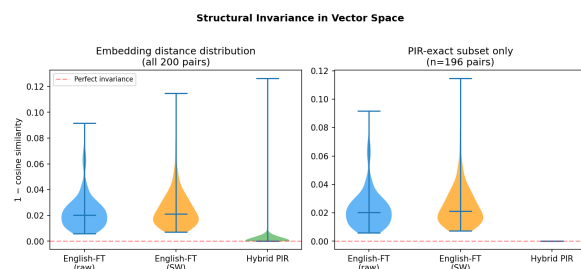


Figure 4: Violin plots of $1 - \cos$ embedding distance per method. Left: all 200 pairs. Right: the 196 PIR-exact pairs only. Hybrid PIR collapses to 0.0000 on exact pairs by construction; English-FT retains 0.0229 mean distance on the same pairs.

tural guarantee even on the pairs where PIR canonicalizes correctly. On the 4 PIR-inexact pairs, Hybrid PIR distance rises sharply to 0.0381, cleanly separating the parser-success from parser-failure regimes. Figure 4 shows the full per-pair distribution split by PIR-exact status.

6 Limitations

Parser accuracy bounds the guarantee. The invariance guarantee is a property of the compiler given correct dependency parses, not of parsing itself. `en_core_web_sm` is trained on general-domain treebanks and achieves 0% exact invariance on biomedical text, where technical terms are frequently misidentified as root verbs and Latinate passive constructions are parsed inconsistently. Deploying PIR in technical domains would require a domain-adapted parser or targeted rule additions for the specific failure modes. Parse cost (~ 4 ms/sentence on CPU) also makes real-time query encoding impractical; the intended deployment mode is offline corpus indexing, where the corpus is pre-parsed once and the cost is amortized over many queries. Importantly, PIR does not eliminate uncertainty but *converts* it: neural similarity failures can be difficult to diagnose, whereas parser failures are structural, domain-typed, and detectable from the dependency graph (e.g., by checking whether the root verb was found), making them easier to audit and correct. For biomedical, legal, or otherwise specialized deployment, a domain-adapted UD parser (e.g., `scispaCy`) can be substituted with no compiler change, since the UD $\rightarrow$ *kāraka* mapping is parser-agnostic.

Information loss limits retrieval quality. Syntactic canonicalization discards surface signals that dense retrieval models rely on for exact matching:

spatial prepositions collapse to a single .LOC role, definite articles and auxiliary verbs are dropped, and exact entity surface forms may be lemmatized away. The Hybrid PIR deduplication mechanism recovers part of this signal by restoring content lemmas absent from the role structure, but a 20-point nDCG@10 gap relative to the strongest retrieval-specialized baseline (BGE-small, .720) on BEIR SciFact remains. This residual gap is most pronounced for scientific and technical queries, where precise keyword overlap is a strong retrieval signal. Some information loss is an inherent consequence of any structural abstraction, and the appropriate trade-off depends on the application: for corpora where syntactic paraphrase variation is high, the canonicalization benefit outweighs the surface signal cost.

Scope of the invariance guarantee. The guarantee applies specifically to syntactic alternations that preserve propositional content: active/passive voice, dative shift, and topicalization. PIR is not designed for adversarial settings where word-scrambling changes agent-patient assignments; in such cases the compiler correctly produces *distinct* role strings for different role assignments, but the retrieval-trained model is not explicitly optimized to amplify those structural differences. Extending the guarantee to a broader class of meaning-preserving transformations — such as nominalization, clausal embedding, or cross-sentential coreference — remains an open direction for future work.

Locative role collision. Of the six classical kārakas, the compiler’s schema covers five (Agent, Object, Dative, Instrumental, Locative); Modifier is a non-kāraka addition. Testing confirmed that source/destination location pairs (e.g. *from X to Y*) do not collapse in practice, since *to*-destinations fold into Dative rather than Locative and *from*-sources are dropped entirely; but a sentence with two genuine locatives (via *in/on/at*) tags both .LOC and sorts them alphabetically, so *The manager held the meeting in Delhi about the deal in Mumbai* collapses with the two cities swapped. None of the 200 diagnostic pairs exhibit this construction, so reported invariance numbers are unaffected; a dedicated .SRC tag and clause-scoped locative tracking are natural future work, alongside the extension in Appendix H.

7 Conclusion

We introduced the Pāṇinian Intermediate Representation (PIR), a deterministic compiler that provides syntactic invariance by construction: active and passive forms of the same proposition map to identical strings whenever the dependency parser assigns correct structure, a guarantee that learned encoders cannot provide. A 200-pair diagnostic suite confirms this behavior on clean general-domain text (196/200 exact matches); the four failures are structurally typed and traceable to known parser limitations. Stopword removal, by contrast, produces distinct strings for all 200 pairs.

To make PIR practical for downstream tasks, we introduced Hybrid PIR, which supplements the role-tagged prefix with a deduplicated set of content lemmas absent from the role structure. This design recovers 58% of the similarity gap between PIR-MiniLM (SNLI) (0.756) and natural language (English, no FT; 0.854), matching stopword removal on STS-B and SICK-R (0.813 vs. 0.815 avg. ρ ; the difference is not statistically significant, see §4.2) at 43.8% token compression, while reaching 80% of the English (no FT) baseline on BEIR SciFact at 39% compression. The three-variant ablation confirms that deduplication is load-bearing: replacing it with unrestricted lemma appending degrades BEIR by 3.5 points and turns the compression benefit negative (−3.0% token compression, Table 2), validating the specific design choice rather than the general idea.

Our QQP duplicate-detection experiment (§4.6) validates the intended positioning: as a deterministic canonicalization tool for offline deduplication, canonical-index construction, and cache keys, PIR delivers precision comparable to a tuned embedding threshold with no tuning and no per-pair encoding cost. It is not, and does not aim to be, a substitute for dense retrieval on pragmatics-sensitive tasks.

The central open challenge is domain generalization. The invariance guarantee is bounded by parser accuracy: biomedical and highly technical domains remain out of reach for the current general-domain model. Future work should investigate domain-adapted parsers, multilingual rule extensions, and contrastive training objectives that explicitly reward structural discrimination beyond the syntactic alternations characterized here.

Acknowledgments

AI assistants (Claude, via Claude Code) were used during the camera-ready revision to help with writing and grammar, and to run and debug evaluation code. All experimental design, the PIR methodology, and the paper’s conclusions are the authors’ own.

References

- Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2016. [MS MARCO: A human generated machine reading comprehension dataset](#). In *Proceedings of the Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches (CoCo@NIPS 2016)*.
- Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.
- Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal. 1995. *Natural Language Processing: A Paninian Perspective*. Prentice Hall India, New Delhi.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. [A large annotated corpus for learning natural language inference](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. Association for Computational Linguistics.
- Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. [SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation](#). In *Proceedings of the 11th International Workshop on Semantic Evaluation*, pages 1–14. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186. Association for Computational Linguistics.
- William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, pages 9–16.
- Katja Filippova and Michael Strube. 2008. Dependency tree based sentence compression. In *Proceedings of the Fifth International Natural Language Generation Conference*, pages 25–32. Association for Computational Linguistics.
- Luyu Gao and Jamie Callan. 2021. Condenser: a pre-training architecture for dense retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 981–993. Association for Computational Linguistics.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910. Association for Computational Linguistics.
- Matthew Henderson, Rami Al-Rfou, Brian Strope, Yunhsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. In *arXiv preprint arXiv:1705.00652*.
- Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. [spaCy: Industrial-strength natural language processing in Python](#). *Zenodo*.
- Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3543–3556. Association for Computational Linguistics.
- Hervé Jégou, Matthijs Douze, and Cordelia Schmid. 2011. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128.
- Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. LLMingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376. Association for Computational Linguistics.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.
- Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781. Association for Computational Linguistics.
- Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the*

- 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 39–48.
- Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 216–223.
- Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-T5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874. Association for Computational Linguistics.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*, pages 1659–1666.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Peng Shi and Jimmy Lin. 2019. Simple BERT models for relation extraction and semantic role labeling. In *arXiv preprint arXiv:1904.05255*.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601. Association for Computational Linguistics.
- Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Ashwini Vaidya, Samar Husain, Prashanth Mannem, and Dipti Misra Sharma. 2009. [A karaka based annotation scheme for English](#). In *Computational Linguistics and Intelligent Text Processing (CICLing 2009)*, volume 5449 of *Lecture Notes in Computer Science*, pages 41–52. Springer.
- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.
- David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550. Association for Computational Linguistics.
- Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.
- Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 5776–5788.
- Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muenighoff, Defu Lian, and Jian-Yun Nie. 2024. [C-pack: Packed resources for general Chinese embeddings](#). In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 641–649.
- Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2024. RECOMP: Improving retrieval-augmented LMs with context compression and selective augmentation. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*.
- Yuan Zhang, Jason Baldridge, and Luheng He. 2019. [PAWS: Paraphrase adversaries from word scrambling](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308. Association for Computational Linguistics.

A Hyperparameters and Training Details

Table 7 lists the training configuration shared across all PIR models and the English-FT (SNLI) baseline. All hyperparameters were fixed before any evaluation; no per-model tuning was performed. We use a temperature of $\tau = 0.05$, fixed before evaluation, and FP16 mixed precision was applied throughout without affecting final metrics.

B Detailed Qualitative Examples

Table 8 illustrates the transformation of diverse English sentences into Pure PIR and Hybrid PIR formats. The examples cover entity merging (*Ronald Reagan* $\rightarrow$ *ronald_reagan.AGT*), active/passive canonicalization (row 4), and modifier recovery via

Parameter	Value
Optimizer	AdamW
Learning Rate	2e-5
Weight Decay	0.01
Warmup Ratio	0.10
Batch Size	128
Max Sequence Length	128 tokens
Temperature (τ)	0.05
Precision	Mixed (FP16)

Table 7: Training hyperparameters shared across all models.

the lexical tail (*foreign, lazy, powerful*). The deduction rule is visible in row 4: because *strategy* and *ceo* are already represented as .OBJ and .AGT respectively, the Hybrid output is identical to Pure PIR.

C Complete Similarity Results

Tables 9 and 10 report the full STS-B and SICK-R Pearson ρ values for all systems. Retrieval-specialized baselines (E5-small, BGE-small) score within 0.017 of English-FT (SNLI) on average, confirming that models of comparable size converge to similar STS performance regardless of training objective. Among PIR variants, the largest single jump is from PIR (no FT) to PIR-MiniLM (SNLI) (+0.093 avg. ρ), showing that representational fine-tuning rather than the compiler drives most of the similarity recovery.

D Compiler Ablation Study

Table 11 reports the PIR compiler component ablation, evaluating progressively richer role assignment rules on SNLI-trained PIR-MiniLM. V1 uses only basic subject-object-root relations; V2 adds named entity preservation via multi-word token merging. The decisive gain comes from V3, which adds the full role inventory (.DAT, .INS, .LOC, .MOD) and tense normalization, improving average ρ by 0.020 over V2 and confirming that Pāṇinian role granularity is the key driver of similarity recovery.

E Extended Invariance Results

Table 12 breaks down the full 261-pair invariance evaluation. The 200-pair diagnostic suite (top section) consists of templated syntactic alternations drawn from clean general-domain text and is the primary evidence discussed in the paper. The 61 stress-test pairs isolate known parser failure modes:

21 biomedical sentences (domain shift) and 40 mined SNLI pairs (multi-clause complexity). The 0/61 rate on the stress test is expected and documents the boundary of the general-domain parser rather than a compiler failure; domain-specific parsers or targeted rule patches would address each failure class independently.

F Attention Sparsification

Figure 5 provides a qualitative view of how PIR redistributes encoder attention. The English baseline diffuses attention across syntactic function words such as “was” and “by,” which carry no semantic content but vary between active and passive forms. PIR-MiniLM receives an identical canonical string for both surface forms, concentrating attention mass on the semantic participants (*ceo*.AGT, *strategy*.OBJ). This sparsification is a direct consequence of the structural canonicalization, not a learned behavior. Note that attention weights are not a causal saliency measure (Jain and Wallace, 2019); this figure illustrates the structural sparsification effect rather than making a causal claim.

G Natural Paraphrase Invariance: Full Results

Table 13 reports the full breakdown of the natural-paraphrase evaluation cited in §4.4. We use PAWS-Wiki (labeled_final, positive label only, 28,904 pairs) and MRPC (glue/mrpc, positive label only, 2,753 pairs) for a combined 31,657 naturally-occurring paraphrase pairs. Because natural paraphrases include heavy lexical substitution (*said* vs. *stated*, entity-name variants, sub-clausal reordering) beyond the syntactic-alternation class the compiler targets, we report both exact string identity (the target of PIR’s structural guarantee) and near-invariance (token Jaccard ≥ 0.8).

H Extended-Schema Probe: Negation, Modality, Quantification

The main-text PIR compiler assigns five kāraka-derived roles (Agent, Object, Dative, Instrumental, Locative) plus a non-kāraka Modifier role and an Action label. Since submission, we prototyped a schema extension that adds three role suffixes, .NEG (negation scope), .MDL (modal auxiliary), and .QNT (quantifier), to test whether the compiler generalizes to alternations that hinge on operator scope rather than argument permutation.

Original English Sentence	Pure PIR Output	Hybrid PIR Output
The researcher analyzed the data with a powerful computer.	researcher.AGT data.OBJ computer.INS ANALYZE.ACT.PAST	researcher.AGT data.OBJ computer.INS ANALYZE.ACT.PAST powerful
A quick brown fox jumped over the lazy dog.	fox.AGT dog.OBJ quick.MOD brown.MOD jump.ACT.PAST	fox.AGT dog.OBJ quick.MOD brown.MOD jump.ACT.PAST lazy
Ronald Reagan implemented foreign policies in Washington.	ronald_reagan.AGT policy.OBJ washington.LOC IMPLEMENT.ACT.PAST	ronald_reagan.AGT policy.OBJ washington.LOC IMPLEMENT.ACT.PAST foreign
The strategy was implemented by the CEO.	ceo.AGT strategy.OBJ IMPLEMENT.ACT.PAST	ceo.AGT strategy.OBJ IMPLEMENT.ACT.PAST

Table 8: Four hand-selected illustrative examples of the PIR compilation process (not a random sample). In the final example, the Hybrid output is identical to Pure PIR because all content lemmas are already represented in the role tags.

System	STS-B	SICK-R	Avg	Cmp
<i>Retrieval-specialized baselines</i>				
English (no FT)	.870	.839	.854	0.0%
English-FT (SNLI)	.860	.823	.841	0.0%
E5-small	.867	.839	.853	0.0%
BGE-small	.883	.832	.858	0.0%
<i>Deterministic compression baselines</i>				
SW	.857	.773	.815	48.3%
SW+L	.849	.761	.805	48.3%
DepLin	.672	.679	.675	65.0%

Table 9: Baseline similarity results and compression ratios.

System	STS-B	SICK-R	Avg	Cmp
<i>PIR variants (ours)</i>				
PIR (no FT)	.641	.684	.663	51.6%
PIR-MiniLM (SNLI)	.753	.760	.756	51.6%
PIR-MiniLM (MARCO)	.755	.767	.761	51.6%
PIR-MiniLM (Hybrid)	.840	.786	.813	43.8%
PIR-MiniLM (Lexical)	.836	.780	.808	3.7%

Table 10: PIR-MiniLM similarity results across compiler variants.

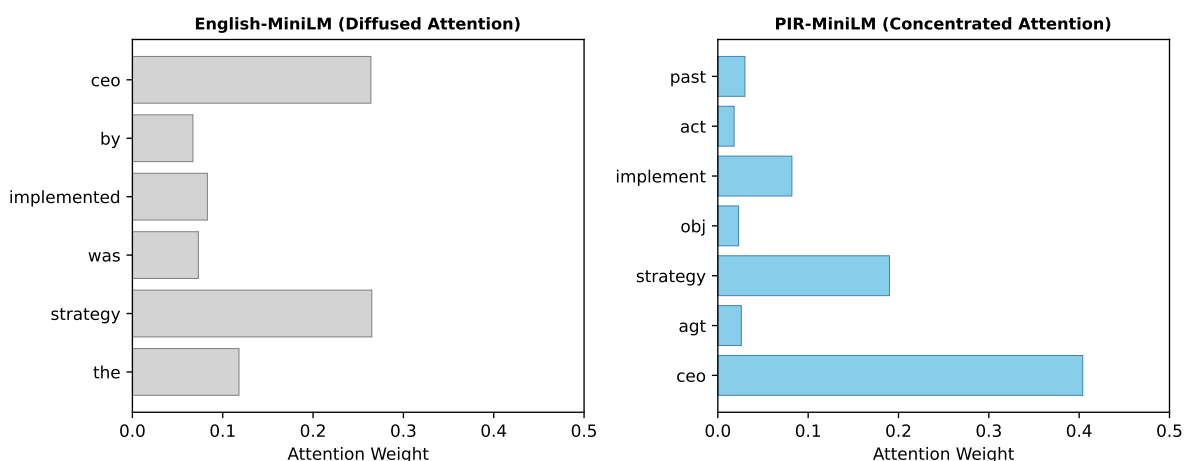


Figure 5: CLS-to-token attention weights in the final transformer layer (mean over 12 heads). Left two panels: English baseline assigns different distributions to the active and passive forms. Right two panels: PIR-MiniLM receives identical strings and produces identical attention distributions, concentrating mass on ceo.AGT (0.404) and strategy.OBJ (0.190).

Configuration	STS-B	SICK-R	Avg
V1: Basic roles (nsubj, dobj, root)	.725	.744	.735
V2: + Entity preservation	.725	.747	.736
V3: Full PIR (all roles + tense)	.753	.760	.756

Table 11: PIR compiler ablation (SNLI-trained PIR-MiniLM). Adding .DAT, .INS, .LOC, and .MOD roles alongside tense normalization provides the largest gain.

Category	<i>n</i>	PIR id./ <i>n</i>	SW id./ <i>n</i>
Diagnostic Suite	200	196/200	0/200
Active/Passive	135	133/135	0/135
Dative Shift	30	28/30	0/30
Topicalization	35	35/35	0/35
Stress Test (OOD)	61	0/61	0/61
Biomedical (OOD)	21	0/21	0/21
Mined SNLI (Noise)	40	0/40	0/40
Total	261	196/261	0/261

Table 12: Full structural invariance results. SW achieved 0 exact-identity matches across all 261 pairs.

Method	Combined		PAWS-Wiki	
	Exact	Near	Exact	Near
Raw	0.19%	86.49%	0.20%	94.19%
SW	12.55%	73.56%	13.73%	79.71%
Base PIR	13.97%	31.45%	15.23%	33.89%
Hybrid PIR	8.52%	28.61%	9.31%	31.00%

Table 13: Natural-paraphrase invariance on PAWS-Wiki + MRPC positive pairs ($n=31,657$ combined). PIR wins on exact-string identity (the target of the compiler’s structural guarantee); SW wins on near-Jaccard because it preserves all content lemmas rather than compressing them into role slots. Hybrid PIR trades exact-string identity for retrieval quality (§4.4), the same tradeoff observed on templated data.

Table 14 reports results on a 60-pair diagnostic probe partitioned into three groups (NEG / MDL / QNT), each with 10 meaning-preserving pairs (active $\leftrightarrow$ passive with the operator preserved) and 10 meaning-changing pairs (operator inserted, negated, or scope-shifted).

The probe here documents that the kāraka role framework generalizes to operator-scope alternations without architectural change.

Group	<i>n</i> each	Base PIR	Extended
NEG (invariance)	10	0%	100%
NEG (discrim.)	10	100%	100%
MDL (invariance)	10	0%	100%
MDL (discrim.)	10	100%	100%
QNT (invariance)	10	100%	100%
QNT (discrim.)	10	50%	100%
Overall	60	58.3%	100.0%

Table 14: Extended-schema probe. Base PIR collapses same-meaning negation and modal pairs into the wrong equivalence class because a token-level dropped operator changes the string; adding .NEG, .MDL, and .QNT role suffixes both preserves invariance under argument permutation and discriminates operator-scope differences. The extension is additive (compiler-only, no retraining) and does not affect main-text results, which are reported under the base six-role schema.