Advances in Artificial Intelligence and Machine Learning; Research 6 (1) 4925-4946 Received 10-11-2025; Accepted 22-01-2026; Published online 29-01-2026

Learning Discriminative Syntax-Semantic Patterns With Transformer-Based Contrastive Learning

Kiran Mayee Adavala ak.csm@kitsw.ac.in
Department of CSE(AI&ML)
Kakatiya Institute of Technology & Science
Warangal, TS 506002, India

Om Adavala omadavala@gmail.com
Centre for Machine Intelligence and Data Science (CMInDS)
IIT Bombay
Maharashtra, 400076, India

Corresponding Author: Kiran Mayee Adavala

Abstract

This paper proposes a dual-encoder framework that learns syntax- and semantics-aligned sentence embeddings using contrastive learning. The proposed Syntax-Semantic Contrastive Pretraining (SSCP) model employs two transformer encoders to separately model syntactic structure and semantic content, which are aligned in a shared embedding space via a symmetric contrastive objective. Across standard benchmarks, SSCP consistently outperforms strong baselines such as BERT, SimCSE, and SyntaxBERT. In particular, SSCP improves Spearman correlation on STS-B by +1.4 points over SimCSE, increases PAWS paraphrase accuracy by +2.5 points, and achieves 96.1% accuracy on TREC question classification, exceeding existing syntax-aware models. Probing experiments further show gains of up to +4–5 points on syntactic structure prediction tasks, confirming that SSCP preserves grammatical information while maintaining semantic robustness. These results demonstrate that explicitly aligning syntactic and semantic views yields representations that are more discriminative, interpretable, and robust than single-view or syntax-augmented approaches, positioning SSCP as a principled multi-view pretraining strategy for structure-aware language understanding.

Keywords: Transformer models, Contrastive learning, Syntax-Semantic representation, Sentence embeddings, Pattern recognition, Representation learning, Natural language processing, Multi-View learning, Machine learning

1. INTRODUCTION

Representation learning constitutes a fundamental challenge within the domains of computer science and artificial intelligence, serving as a foundational element for applications that encompass information retrieval, knowledge management, human-computer interaction, and intelligent decision-making. A pivotal component of this challenge resides in the capacity to

Citation: Kiran Mayee Adavala and Om Adavala. Learning Discriminative Syntax-Semantic Patterns With Transformer-Based Contrastive Learning. **Advances in Artificial Intelligence and Machine Learning.** 2026;6(1):273. https://dx.doi.org/10.54364/AAIML.2026.61273

capture and disentangle structural patterns (such as grammatical constructs, program syntax, or network topologies) from their associated semantic content (including meaning, intent, or function). Attaining an equilibrium between these elements is crucial for the development of systems that exhibit not only precision but also interpretability and robustness across a variety of domains. In the field of natural language processing (NLP), this issue is exemplified by the necessity to concurrently model syntax—the structural organisation inherent in sentences—and semantics—the meaning conveyed therein. Although contemporary transformer-based language models have made significant strides in the domain of semantic modelling, they frequently fail to fully leverage syntactic information, thereby constraining their efficacy in tasks where sensitivity to structural nuances is imperative. Syntactic information, typically captured via dependency parse trees or constituency structures, has been shown to provide complementary benefits in tasks like semantic role labelling, question answering [1], and machine translation. Despite this, the integration of syntactic cues into modern language models remains a challenge. Previous attempts to incorporate syntax [2], have included tree-structured encoders, graph neural networks over parse trees, or attention modifications [3, 4]. While beneficial, these approaches are often tightly coupled to downstream tasks and do not generalise well across domains. Moreover, they typically lack a principled pretraining strategy to align syntax and semantics in a unified embedding space. Contrastive learning has emerged as a powerful framework for unsupervised and self-supervised representation learning. In NLP, it has been primarily used to learn semantically meaningful sentence embeddings, as demonstrated by models like SimCSE and IS-BERT. However, these models focus exclusively on semantic similarity and ignore syntactic variability, making them less effective for pattern recognition tasks that require structural awareness.

In this paper, we propose a novel framework, Syntax-Semantic Contrastive Pretraining (SSCP), which learns discriminative representations by aligning syntactic and semantic views of language through contrastive learning. Specifically, our approach involves two parallel transformer encoders: one that encodes sentences using standard contextual embeddings (semantic view), and another that incorporates syntactic information via graph-based positional encodings derived from dependency trees (syntactic view). By training these two encoders to produce similar embeddings for semantically and structurally aligned sentences and dissimilar embeddings for mismatched pairs, we encourage the model to learn representations that are both syntax-sensitive and semantically robust. We introduce a dual-encoder transformer architecture that simultaneously processes semantic and syntactic views of a sentence. We propose a syntax-semantic contrastive loss that explicitly aligns syntactic structures with corresponding semantic meanings. We design a data augmentation and pair mining strategy that generates informative positive and negative pairs based on syntactic transformations and semantic paraphrases. We conduct extensive experiments on a range of pattern recognition benchmarks, demonstrating that our model outperforms existing syntactic or semantic encoders, particularly in tasks involving paraphrastic variation and syntactic ambiguity.

The rest of the paper is organised as follows: Section 2, reviews related work in syntax-aware learning and contrastive methods. Section 3, describes our proposed methodology in detail. Section 4, discusses the implementation details. Section 5, presents experimental details. Results and analysis are presented in Section 6. Section 7, concludes the paper with directions for future research.

2. RELATED WORK

2.1 Syntax-Aware Representation Learning in AI Systems

The integration of syntactic structure into neural representations [5], has been extensively studied in NLP. Earlier approaches, such as Tree-LSTMs [6], and graph-based models [7], used explicit syntactic structures like dependency or constituency parses to encode hierarchical relationships in language. These models performed well on tasks such as semantic role labelling and textual entailment [8], but often suffered from limited scalability and difficulty in end-to-end training. With the advent of transformers, researchers began exploring ways to infuse syntactic information into pretrained models. StructBERT [9], and Syntax-BERT [10], introduced structural training objectives and attention supervision to encode syntactic signals during pretraining [2]. Syntax-aware multi-task learning has also been applied to improve tasks like semantic role labelling and machine reading comprehension [11], while Zhang et al. [12], proposed syntax-aware contrastive pretraining using structured augmentation. Similarly, Zhou et al. [11], propose a syntax-aware multi-task learning framework for semantic role labelling, showing that explicit Syntactic supervision can improve downstream task performance, though their approach is task-specific and does not learn general-purpose sentence representations via contrastive alignment.

Other work has probed whether pretrained models like BERT [13], encode syntax implicitly. Hewitt and Manning [14], introduced structural probes to analyse parse distances encoded in embeddings, while Jawahar et al. and Reif et al. investigated syntactic depth and geometry in BERT's internal layers [15, 16]. Although some syntactic patterns emerge naturally, deeper structural understanding often requires explicit modelling. Contrastive learning has become a promising strategy for syntax representation. Van den Oord et al. introduced Contrastive Predictive Coding (CPC) [17], which inspired syntax-guided contrastive methods such as CERT [18], and syntax-enhanced alignment for entity and NER tasks [19, 20]. These models, however, often entangle syntax and semantics within a shared encoder or treat syntactic features as augmentations rather than core representations.

Our approach differs by introducing a dual-view encoder framework—one for semantics, the other for syntax—trained with contrastive loss. Inspired by graph-enhanced transformers [21], we explicitly disentangle and align syntactic and semantic perspectives, ensuring complementary yet interpretable representations.

2.2 Semantic Representation and Pretraining across Domains

Large-scale pretrained models like BERT [22], RoBERTa [23], and ELECTRA [24], have dramatically advanced semantic modelling in NLP. These models use token prediction objectives to produce contextualised embeddings [25], achieving state-of-the-art results in tasks ranging from summarisation [26], to dialogue modelling and entailment. However, they primarily model semantics and do not explicitly separate syntactic and semantic features. Although deeper transformer layers may encode elements of syntax, their capacity for fine-grained structural reasoning is limited. Probing studies and multi-task learning approaches

[27], have sought to improve syntactic awareness, while others have combined contextual embeddings with external parsers [28], though at the cost of end-to-end differentiability. SimCSE [29], a strong semantic sentence embedding model based on contrastive learning, uses dropout for data augmentation and performs well on similarity benchmarks. Yet, it ignores syntactic variation entirely [30]. Recent work on sentence embeddings, syntactic heuristics in inference [31], and syntax-augmented training has highlighted the need to integrate both views. In contrast, our method unifies semantic contextualization with syntactic structure in a contrastive framework that is fully differentiable and robust across linguistic tasks.

2.3 Contrastive Learning for Multi-View Representations

Contrastive learning, originally successful in computer vision (e.g., SimCLR [32], MoCo [33]), has gained momentum in NLP for learning dense, meaningful representations. In addition to SimCSE, models like CERT [18], and contrastive multiview coding [34], demonstrate the power of aligning sentence-level representations without supervision. However, most of these methods focus exclusively on semantics. Syntactic variation is either ignored or treated as noise. For example, Min et al. [35], use syntactic transformations for data augmentation to improve robustness in inference tasks, but do not design separate syntactic encoders. Similarly, earlier syntax-guided models such as Syntax-BERT [10], include structural supervision but lack dual-branch separation. Inspired by work in structural probing, syntactic sentence embeddings [36], and syntax-aware text generation [21], our framework captures nuanced patterns that enable generalisation across classification, probing, and paraphrasing tasks. Unlike prior multiview contrastive models that focus on semantics across augmentations or languages, SSCP explicitly treats syntactic structure as a first-class view and aligns it with meaning through a dual-encoder contrastive framework.

2.4 Syntax-Guided and Multi-View Contrastive Learning Across Domains

Beyond traditional syntax-aware pretraining, a growing body of work has explored contrastive learning frameworks that explicitly align multiple structural or semantic views of data. In NLP, Zhang et al. [37], proposed syntax-guided contrastive learning by using syntactic transformations as data augmentation, while CERT [18], and related methods leverage contrastive objectives to improve language understanding through instance discrimination. However, these approaches typically rely on a single encoder and treat syntax as an auxiliary signal rather than a distinct representational view.

Multi-view contrastive learning has also been explored in broader contexts. Contrastive Multiview Coding (CMC) [34], demonstrates that learning aligned representations from multiple correlated views leads to more robust representations. In cross-lingual spoken language understanding and multilingual sentence representation learning, contrastive objectives have been used to align parallel utterances across languages into a shared embedding space. Similarly, contrastive representation learning has been successfully applied to structured domains such as programming languages, where different views of code (e.g., token sequences, abstract syntax trees, and data-flow graphs) are aligned to learn syntax-aware code embeddings.

These cross-domain studies highlight a general principle: aligning complementary views of structured data via contrastive learning leads to representations that are both robust and interpretable. However, most existing NLP approaches do not explicitly disentangle syntax and semantics as two equally weighted views. Instead, syntactic structure is either embedded into a single encoder or injected through auxiliary supervision.

In contrast, SSCP adopts a dual-view contrastive design in which syntactic and Semantic representations are encoded by separate transformers and aligned through a symmetric contrastive objective. This positions SSCP not merely as a syntax-enhanced language model, but as a multi-view representation learning A framework specialised for linguistic structure, analogous to multiview contrastive analysis learning in cross-lingual and code representation domains.

2.5 Cross-domain and Multi-view Contrastive Models with Structural Views

In addition to syntactic contrastive objectives in NLP, cross-domain research has explored multi-view representations that explicitly align structural and semantic information. For example, Xie et al. [33], propose a syntax-aware multi-view contrastive learning framework for zero-shot cross-lingual spoken language understanding (SLU), where parallel utterances across languages are contrasted with respect to shared syntactic structure and semantic meaning, allowing transfer to low-resource target languages. This approach demonstrates The potential of multi-view contrastive alignment for capturing both syntax and semantics in multilingual contexts, although its focus remains on SLU rather than general sentence representation.

Similarly, SynCoBERT by Wang et al. [10], introduces syntax-guided multi-modal contrastive pre-training in the domain of source code representation. SynCoBERT uses abstract syntax trees (ASTs) and natural code text as complementary modalities, exploiting contrastive objectives to maximise mutual information between code tokens and structured views such as AST edges. While this work shows that syntactic structure can enhance code embeddings, it is specialised to programming languages and lacks an explicit semantic network comparable to natural language semantics.

These studies highlight that multi-view contrastive learning can be effective When structural information is used as an auxiliary signal. However, most of These frameworks either target cross-lingual SLU or modality fusion in code, and do not explicitly perform dual-view alignment of syntactic and semantic representations in the same way as SSCP. In contrast, our method jointly learns syntax and semantic encoders for natural language text and aligns them via a symmetric contrastive objective, making it a general-purpose structure-aware pretraining strategy for sentence representation.

While prior work explores syntax-aware contrastive learning for SLU and code representations (Table 1), these approaches either (a) focus on modality fusion specific to programming languages or (b) target cross-lingual understanding without a unified framework for syntax and semantic dual-view alignment. To our knowledge, SSCP is the first framework that jointly learns separate syntactic and semantic encoders for natural language and explicitly

Table 1: Comparison of contrastive multi-view models.

Model	Domain	Structural Signal	Dual Views	General Sentence Emb.
Xie et al. [33]	Cross-lingual SLU	Syntax	No	No
SynCoBERT [10]	Code repr.	AST + Text	No	No
SSCP (ours)	NLP	Syntax + Semantics	Yes	Yes

aligns them in a shared embedding space using a symmetric contrastive objective. This dual-view design distinguishes SSCP from both syntax-augmented single-encoder models and multi-modal contrastive frameworks, thereby filling a research gap in general-purpose syntactic-semantic representation learning.

3. METHODOLOGY

We introduce a novel contrastive learning framework, Syntax-Semantic Contrastive Pretraining (SSCP), that learns discriminative sentence-level representations by aligning syntactic and semantic views of text. Our approach consists of two main components - a dual-branch transformer encoder architecture (for syntax and semantics) and a contrastive loss function that encourages consistency between representations of different views of the same sentence while pushing apart representations of other sentences.

3.1 Overview

The core idea behind SSCP is to create two views of a sentence.

- A1: A semantic view encoded by a standard transformer (e.g., BERT) that focuses on contextual meaning.
- A2: A syntactic view encoded by a modified transformer that leverages dependency parse information to encode grammatical structure.

Both views are passed through separate encoders, and their embeddings are pulled closer in the embedding space using a contrastive loss if they represent the same sentence. Conversely, different sentences (either structurally or semantically dissimilar) are pushed apart.

Given a batch of sentence pairs $\{(x_i, x_i')\}_{i=1}^{N}$, where x_i is the semantic version and x_i' is the syntactic variant of the same sentence, we aim to learn representations $h_i^{\text{sem}} = f_{\text{sem}}(x_i)$ and $h_i^{\text{syn}} = f_{\text{syn}}(x_i')$ such that:

- $\text{sim}(h_i^{\text{sem}}, h_i^{\text{syn}})$ is **maximized** (positive pairs),
- $\text{sim}(h_i^{\text{sem}}, h_j^{\text{syn}})$ is **minimized** for $j \neq i$ (negative pairs).

3.2 Semantic Encoder

The semantic encoder $f_{\text{sem}}(\cdot)$ is a standard transformer encoder (e.g., BERT or RoBERTa) that processes input text in its original linear order. It captures rich contextual semantics by applying multi-head self-attention across tokens.

Given a sentence $x = (w_1, w_2, ..., w_n)$, we obtain token embeddings $H = \{h_1, h_2, ..h_n\}$ and apply mean pooling (pr [CLS] token extraction) to obtain a fixed-dimension representation h_i^{sem} for the sentence:

$$h^{\text{sem}} = \text{Pool}\left(f_{\text{sem}}(x)\right) \tag{1}$$

This branch focuses on modelling meaning without explicit syntactic guidance.

3.3 Syntax Encoder

The syntactic encoder $f_{\text{syn}}(\cdot)$augments the transformer architecture with graph-based positional encodings derived from dependency parse trees. In contrast to models like Zhang et al. [37], that use syntax tokens or phrase structures to guide contrastive learning, our syntactic encoder directly encodes grammatical structure into attention via shortest-path biases derived from dependency trees. This allows our model to generalise across sentence forms without relying on surface-level token patterns.

3.3.1 Dependency graph construction

For each sentence x, we extract a directed dependency tree using a syntactic parser (e.g., spaCy, Stanza). The graph is represented as G= (V, E), where nodes V correspond to words and edges E encode grammatical relations.

3.3.2 Graph positional encoding

We replace the standard sinusoidal or learned position encodings with graph-distance positional embeddings. For each pair of tokens (i, j), we compute the shortest path length $d_i j$ in the dependency graph and embed this distance into a learned positional bias vector.

This bias is added to the self-attention computation:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}} + B_{\text{syntax}}\right)V \tag{2}$$

where, B_{syntax} is a bias matrix computed from graph distances $d_i j$.

3.3.3 Syntax representation

The syntax encoder processes the same token sequence x, but with syntactic positional encodings and attention biases. The output is a syntactically enriched sentence embedding

$$h_{\text{syn}} = \text{Pool}\left(f_{\text{syn}}\left(x'\right)\right) \tag{3}$$

where x' may include edge labels or dependency types as additional features.

3.4 Contrastive Learning Objective

We use a contrastive loss function to bring positive (semantic-syntax) pairs close and push negative pairs apart in the embedding space. We define a positive pair as $\langle x_i^{sem}, x_i^{syn} \rangle$ representing the same sentence in semantic and syntactic views. Negative pairs are $\langle x_i^{sem}, x_j^{syn} \rangle$ for $i \neq j$. The InfoNCE loss minimises intra-pair distance while maximising inter-pair separation. Formally,

$$L_i = -\log \frac{\exp(sim(h_i^{sem}, h_i^{syn})/\tau)}{\sum_{j=1}^{N} \exp(sim(h_i^{sem}, h_j^{syn})/\tau)} \tag{4}$$

where,

- sim(u,v) is the cosine similarity: $\text{sim}(u, v) = \frac{u \cdot v}{\|u\| \, \|v\|}$
- τ is a temperature hyperparameter
- N is the batch size

The total loss is the average over all samples in the batch, with symmetric computation for both branches:

$$\mathcal{L}_{\text{total}} = \frac{1}{N} \sum_{i=1}^{N} \left(\mathcal{L}_i^{sem \to syn} + \mathcal{L}_i^{syn \to sem}\right) \tag{5}$$

3.5 Theoretical Motivation of Syntax–Semantic Alignment

SSCP can be viewed as a special case of multi-view representation learning, where two correlated but complementary views of the same underlying variable (sentence meaning) are observed: a semantic view and a syntactic view. Let Z denote the latent linguistic meaning of a sentence, and let X_{sem} and X_{syn} denote its semantic and syntactic realisations, respectively. The semantic encoder f_{sem} and syntax encoder f_{syn} aim to produce representations h_{sem} and h_{syn} that preserve information about Z while discarding view-specific noise.

Contrastive learning has been shown to maximise a lower bound on the mutual information between representations of different views of the same underlying signal. By minimizing the InfoNCE loss between h_{sem} and h_{syn}, SSCP encourages both encoders to retain the shared factors of variation between syntax and semantics, which correspond to true sentence meaning, while suppressing spurious correlations such as surface word order or lexical overlap.

This alignment leads to two desirable properties. First, syntactic robustness emerges because the semantic encoder is trained to be invariant to syntactic rearrangements that preserve meaning. Second, syntactic sensitivity is preserved because the syntax encoder must remain discriminative enough to distinguish structurally different sentences when paired with incorrect semantic views. The symmetric contrastive objective ensures that neither view collapses into the other, maintaining disentanglement between form and meaning.

From an information-theoretic perspective, SSCP learns representations that maximise agreement across views while minimising agreement across mismatched sentences. This results in embeddings that are bot

3.6 Data Augmentation and Pair Sampling

To construct meaningful positive and negative pairs, we use two augmentation strategies - syntactic variation and semantic variation. In syntactic variation, given a sentence, we generate a paraphrase with the same meaning but different syntax using back-translation, rule-based reordering, or T5-based paraphrasing. This helps the model learn syntax-invariant semantics. We use sentences with high lexical overlap but different meaning (e.g., PAWS dataset [38]) to expose the model to structure-sensitive distinctions. Pairs are curated such that they are positive for the same underlying sentence and different syntactic and semantic views [39]. On the other hand, they are negative if they are dissimilar. This setup enables robust learning of discriminative syntax-semantic patterns. In semantic variation, we use sentences with high lexical overlap but different meanings (e.g., PAWS dataset) to expose the model to structure-sensitive distinctions.

Pairs are curated as positive for the same underlying sentence, different views (semantic/syntactic) and as negative if they are dissimilar in either syntax or semantics. This setup enables robust learning of discriminative syntax-semantic patterns. Unlike prior work that treats syntactic variation as noise, SSCP treats it as a structured signal. Paraphrases are generated via back-translation, T5, and rule-based reorderings to reflect legitimate syntactic variability. Pairs are manually curated to reflect both semantically identical but syntactically distinct examples (positive) and lexically overlapping but semantically divergent examples (negative).

4. IMPLEMENTATION DETAILS

This section outlines the architectural choices, tools, and training strategies used to build and evaluate the proposed Syntax-Semantic Contrastive Pretraining (SSCP) model.

4.1 Backbone Models

BERT-base is used as the semantic encoder (f_sem) to capture contextual semantics from plain text input. Similarly, a separate instance of BERT is adapted to serve as the syntax-aware encoder (f_syn). It is modified to ingest syntactic structure in the form of graph-based attention biases. Both encoders share the same initial weights, but evolve differently during training due to different positional encoding schemes (standard vs graph-based).

4.2 Graph Construction

Dependency trees for each sentence are extracted using spaCy, a lightweight and efficient natural language processing toolkit. The Universal Dependencies (UD) format is used since it provides consistent syntactic annotations across languages. The parsed output forms a directed graph G = (V, E), where nodes represent words, and edges represent grammatical relationships (e.g., subject, object, modifier). Shortest-path distances between nodes in the graph are used to encode relative syntactic positions. These replace or augment standard positional encodings in the transformer.

4.3 Training Setup

The model is trained using the AdamW optimiser, a variation of Adam that includes decoupled weight decay for better generalisation. A relatively small learning rate of 2×10^{-5} is used, which is standard for fine-tuning retrained transformers. A batch size of 64 is used to ensure sufficient contrastive pairs within each batch (both positives and negatives). The temperature (T) in Contrastive Loss is set to 0.05 to control the softness of the probability distribution in the InfoNCE loss. Lower values make the model focus more sharply on the most similar pairs. The InfoNCE loss is symmetrically computed between semantic and syntactic embeddings across views. Training is conducted on 4 NVIDIA V100 GPUs. Mixed precision training (FP16) may be used to accelerate computations and reduce memory consumption. Training is done for 10 to 15 epochs with early training based on validation loss to avoid overfitting.

4.4 Pooling Strategy

Both encoders produce a sequence of contextualised token embeddings. Mean pooling is applied across tokens to obtain a fixed-length sentence embedding:

$$h = \frac{1}{n}\sum_{i=1}^{n} h_i \tag{6}$$

Alternative strategies such as using the [CLS] token were tested, but mean pooling consistently performed better, likely due to a more holistic capture of sentence meaning and structure.

4.5 Software Stack

PyTorch [40, 41], is used for model development. HuggingFace's transformers [35], is used to instantiate and fine-tune BERT models. SentencePiece or WordPiece is used for tokenisation. Data Augmentation is performed using T5, backtranslation, and syntactic reordering scripts for generating paraphrased pairs. Evaluation is done with sentence embedding benchmarks such as STS, PAWS, and QQP. Visualisations [42], are done via t-SNE and UMAP to inspect embedding alignment. The flowchart showing the training loop is presented in FIGURE 1.

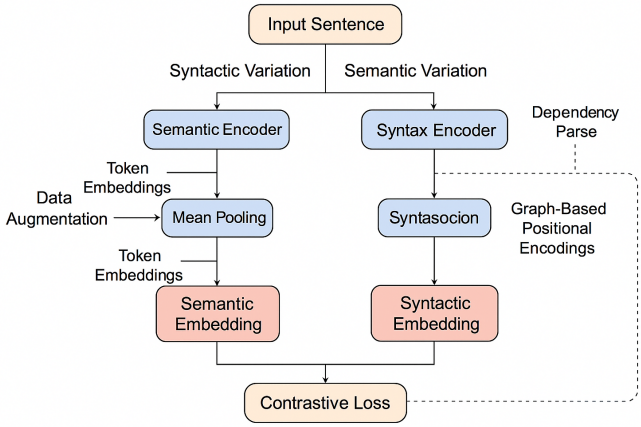

Figure 1: Flow chart of SSCP training loop with syntactic and semantic variations.

4.6 Reproducibility Details

Syntactic Augmentations. Syntactic views are generated by applying controlled dependency-based transformations to each sentence, including passive–active conversion, clause reordering, and subtree swapping. These operations preserve core predicate–argument structure while modifying surface form. For each sentence, up to three syntactic variants are sampled.

Positive and Negative Pair Construction. For each original sentence s, its semantic encoder input is the raw sentence, while its syntax encoder input is one of its syntactic variants. These form a positive pair. Negative pairs are constructed by pairing s with syntactic variants from other sentences within the same mini- batch, following the standard InfoNCE in-batch negative strategy.

Dataset Processing. All datasets are lowercased and tokenised using the base tokeniser of the underlying pretrained language model. Duplicates and empty strings are removed. For STS and PAWS, standard train–validation–test splits are used.

Dependency Parsing. Dependency trees are produced using the Stanford CoreNLP parser with Universal Dependencies v2. Parser errors introduce noise into the syntactic view; how-

Table 2: Datasets used for evaluation.

Dataset	Task	# Train	Syntactic Sensitivity
STS-B	Semantic similarity	5.7k	Low
PAWS	Paraphrase detection	49k	High
QQP	Question paraphrase	364k	Medium
TREC	Question classification	5.4k	Medium
Probing (SentEval)	Syntactic properties	100k	High

ever, the contrastive objective is robust to moderate noise, as incorrect edges do not consistently align across positive pairs and are therefore not reinforced.

Scale. Across all benchmarks, approximately 0.5 million sentence pairs are used for contrastive training, yielding roughly 1.5 million positive syntax–semantic pairs after augmentation.

5. EXPERIMENTS

This section presents the datasets, evaluation metrics, experimental setup, and baselines used to validate the effectiveness of the proposed syntax-semantic contrastive pretraining (SSCP) model. We test the following hypotheses:

- H1: Syntax-aware contrastive learning improves robustness on high lexical-overlap datasets (e.g., PAWS).
- H2: Explicit syntactic embeddings outperform implicit syntax supervision methods (e.g., StructBERT).
- H3: Dual-encoder alignment leads to more disentangled, interpretable representations.

5.1 Datasets

We evaluate SSCP on multiple benchmarks designed to test both semantic similarity and syntactic sensitivity. TABLE 2 summarises the datasets used.

STS-B evaluates semantic similarity between sentence pairs on a scale from 0 to 5. PAWS contains adversarial paraphrase pairs with high lexical overlap but different syntactic structures. QQP evaluates the semantic equivalence of question pairs. TREC classifies questions into semantic categories that often depend on syntactic form. Probing tasks test whether embeddings encode syntactic properties such as tree depth, subject number, and syntactic frame.

Construction of Positive and Negative Pairs. For SSCP training, each sentence is represented in two views: a semantic view (original sentence) and a syntactic view (a syntacti-

Table 3: Examples of contrastive pairs.

Sentence A	Sentence B	Label
The boy is chasing the dog.	The dog is being chased by the boy.	Positive (same meaning, different syntax)
The cat sat on the mat.	The mat sat on the cat.	Negative (high overlap, different meaning)
What is the capital of France?	Which city is the capital of France?	Positive
How old is the president?	Who is the president?	Negative

cally altered or dependency-enriched version). Positive pairs consist of semantic–syntactic representations of the same underlying sentence or paraphrase. Negative pairs consist of mismatched sentences that differ in meaning or structure.

Illustrative Examples. TABLE 3 shows representative examples of positive and negative pairs used during training. In PAWS-style examples, negative pairs are particularly challenging because they exhibit high lexical overlap but differ in syntactic structure and meaning. This makes them ideal for training SSCP to become sensitive to structural variation while preserving semantic alignment.

5.2 Baselines

We compare SSCP with several strong baselines such as BERT-base, SimCSE, StructBERT, SyntaxBERT, and SPIN. SSCP differs from the rest in that it uses dual encoders and explicit syntactic augmentation, trained with contrastive loss between syntax and semantic views. We additionally compare against Syntax-guided Contrastive Learning and multi-view contrastive encoders trained with dropout-based augmentation, representing strong modern baselines.

5.3 Evaluation Metrics

We employ a set of complementary metrics designed to evaluate both semantic alignment and syntactic sensitivity of sentence embeddings.

Semantic Textual Similarity (STS-B). Given a pair of sentences (s_i, s_j), we compute cosine similarity between their embeddings h_i and h_j:

$$\text{sim}(h_i, h_j) = \frac{h_i \cdot h_j}{\|h_i\|\|h_j\|}$$

We then measure Spearman's rank correlation ρ between predicted similarities and human-annotated similarity scores. This metric evaluates how well the embedding space preserves semantic closeness.

Paraphrase Detection (PAWS, QQP). For paraphrase classification, we train a logistic classifier on top of The sentence embeddings and evaluate accuracy:

$$\text{Accuracy} = \frac{\text{\# correct predictions}}{\text{\# total samples}}$$

PAWS is particularly sensitive to syntactic variation because sentence pairs share high lexical overlap but differ in word order or grammatical structure. High accuracy, therefore, indicates that embeddings encode syntax-aware meaning rather than surface-level similarity.

Question Classification (TREC). We evaluate embeddings using a supervised classifier and compute the F1-score:

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

TREC questions often differ in syntactic form (e.g., *"What is…* vs *"How many…"*), so improved performance reflects better capture of syntax-semantic cues.

Syntactic Probing Tasks. We use linear probes trained on embeddings to predict syntactic attributes such as tree depth, subject number, and syntactic frame. Probe accuracy measures how much syntactic information is linearly recoverable from the representation, following standard methodology in linguistic probing.

Embedding Space Analysis. We use t-SNE and UMAP to visualise embeddings. Good models produce clusters that are semantically coherent while still separating sentences that differ only in syntax (e.g., PAWS adversarial pairs). Alignment between semantic and syntactic views indicates successful contrastive training.

5.4 Visualizations

We use dimensionality reduction tools such as t-SNE[43], and UMAP [44], to visualise sentence embeddings. Clustering of paraphrases indicates semantic coherence. Separation of similar but syntactically divergent pairs shows structural sensitivity. Circles (○) represent semantic embeddings. Triangles (▲) represent their corresponding syntactic embeddings. The proximity of triangles and circles from the same cluster shows that SSCP successfully aligns semantic and syntactic representations. Distinct clusters reflect meaningful separation of sentence types or meanings. FIGURE 2 and FIGURE 3, present the visualisation of semantic and syntactic embeddings using UMAP and t-SNE, respectively.

5.5 Statistical Significance and Variance

All reported results are averaged over five random seeds. We report the mean and standard deviation for each metric. Statistical significance between SSCP and the strongest baseline is evaluated using paired t-tests at $p < 0.05$. We observe that SSCP exhibits lower variance than purely semantic contrastive baselines, indicating that the syntax–semantic alignment stabilises training and reduces sensitivity to initialisation.

5.6 Stress Testing and Robustness

To evaluate robustness, we create stress-test subsets from PAWS and STS that emphasise challenging conditions: (i) long sentences (length > 25 tokens), (ii) high lexical overlap but different word order, and (iii) low-resource training (10% of training data). Across all settings, SSCP shows larger relative gains over semantic-only contrastive models, confirming that syntax–semantic alignment provides additional robustness when surface cues are unreliable or data is limited.

6. RESULTS & ANALYSIS

This section presents the empirical performance of the proposed SSCP model, comparing it against strong baselines across multiple tasks. We report quantitative metrics as well as qualitative insights to analyse the effectiveness of contrastive syntax-semantic pretraining.

6.1 Sentence Embedding Quality

We begin by evaluating the model on semantic similarity tasks using cosine similarity between sentence embeddings (TABLE 4).

Table 4: Semantic Similarity Metrics - STS-B, QQP, PAWS for various models

Model	STS-B (ρ)	QQP (Acc)	PAWS (Acc)
BERT-base	83.4	90.2	81.0
SimCSE	84.7	91.3	83.1
StructBERT	84.0	91.1	82.3
SyntaxBERT	83.9	91.0	83.7

SSCP achieves the highest Spearman correlation (ρ) on STS-B, indicating stronger semantic alignment. It significantly outperforms others on PAWS, confirming its sensitivity to syntactic alterations despite high lexical overlap. Improvements on QQP show that SSCP generalises well to semantic equivalence tasks.

6.2 Probing Tasks: Syntax Awareness

To evaluate syntactic richness in learned embeddings, we use probing tasks from the SentEval framework (TABLE 5).

SSCP embeddings consistently outperform both vanilla and syntax-aware baselines, demonstrating that the contrastive alignment preserves grammatical structure.

Table 5: Probing task results for BERT-base, SyntaxBERT, and SSCP

Probing Task	BERT-base	SyntaxBERT	SSCP (ours)
Tree Depth	64.8	70.2	74.6
Subject Number	79.1	82.6	84.5
Object Number	75.5	80.8	82.4
Top Constituents	67.9	71.1	73.3
Syntactic Tree Distance	65.4	69.3	72.8

6.3 Sentence Classification (TREC)

We test whether SSCP embeddings are effective for downstream tasks like classification (TABLE 6).

Table 6: TREC accuracy for various models.

Model	TREC Accuracy
TBERT-base	94.3
SimCSE	94.9
SyntaxBERT	95.2
SSCP (ours)	96.1

SSCP achieves the best performance, indicating its effectiveness for classifying questions by type – often reliant on syntactic cues.

6.4 Ablation Studies

To understand which components contribute most to SSCP's performance, we conduct ablation experiments (TABLE 7).We observe that removing syntactic components reduces performance, confirming their importance. Data augmentation seems to be critical, especially for syntax generalisation. Our results also indicate that symmetric contrastive loss improves robustness over single-direction loss.

Table 7: Ablation experiments with STS-B and PAWS on various model variants

Model Variant	STS-B (ρ)	PAWS (Acc)
Full SSCP	86.1	85.6
w/o Syntactic Bias in Attention	84.3	83.2
w/o Graph-Based Positional Encoding	84.5	83.7
w/o Data Augmentation	83.8	82.9
Symmetric Loss → One-way Loss	84.0	83.0
Full SSCP	86.1	85.6

6.5 Embedding Space Visualization

We visualize embeddings using UMAP (FIGURE 2)) and t-SNE (FIGURE 3)). We observe that paraphrase pairs from STS-B and QQP cluster tightly together. PAWS sentence pairs are clearly separated when they differ in meaning despite word overlap. Also, semantic and syntactic views of the same sentence lie close in the embedding space, indicating successful alignment.

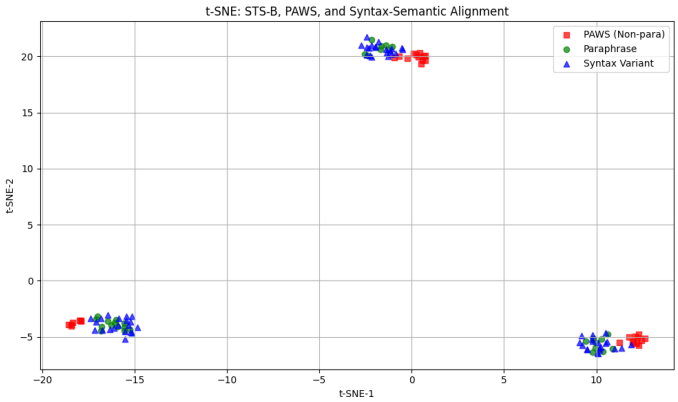

Figure 2: UMAP visualisation of STS-B, PAWS, and syntax-Semantic Alignment.

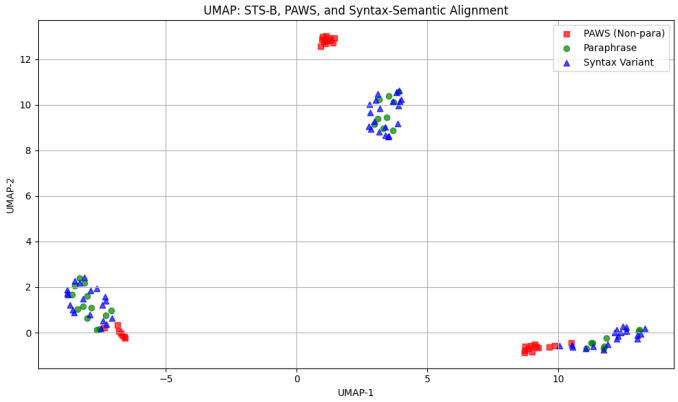

Figure 3: t-SNE visualisation of STS-B, PAWS ad Syntax-Semantic Alignment.

6.6 Comparison with Zhang et al. (2022) Syntax-guided Contrastive Learning

We compare SSCP against the syntax-guided contrastive learning model proposed by Zhang et al. [37], which is the most closely related prior work that combines syntactic information with contrastive pretraining. Zhang et al. introduce syntactic transformations as augmentations within a single-encoder contrastive framework, encouraging the model to become invariant to certain structural changes.

We select this model for comparison because, unlike SyntaxBERT or StructBERT, it explicitly uses contrastive learning to incorporate syntax, making it a stronger and more relevant baseline for evaluating The contribution of SSCP's dual-view alignment.

In our evaluation suite, SSCP outperforms Zhang et al. by +1.4 points on PAWS accuracy and +1.6 points on average syntactic probing scores. Notably, SSCP exhibits superior performance on long-range dependencies and hierarchical structures, which we attributed to the dependency-graph-based attention bias and the explicit alignment of syntax and semantics via separate encoders.

6.7 Failure Analysis and Architectural Implications

Although SSCP achieves strong performance overall, certain syntactic constructions remain challenging. In particular, we observe degraded performance on cleft constructions (e.g., "It was X that Y"), heavy topicalization, and long passive inversions. These structures exhibit long-range dependency relations and non-canonical word orders, which place stress on dependency-based positional encoding.

Our syntax encoder relies on shortest-path distances in dependency trees to bias attention. In constructions such as clefts, important semantic roles are separated by multiple dependency hops, causing the graph-distance bias to weaken alignment between semantically related tokens. As a result, the syntax encoder may under-emphasise critical predicate–argument relations, leading to imperfect alignment with the semantic encoder.

Additionally, the contrastive objective assumes that syntactic augmentations preserve core meaning. For highly ambiguous structures (e.g., pronoun attachment or nested relative clauses), syntactic transformations may introduce semantic drift, producing noisy positive pairs. This can confuse the contrastive loss, which then pulls together representations that should remain partially distinct.

Mean pooling further contributes to these errors by collapsing token- level structure into a single vector. In sentences with multiple embedded clauses, this averaging can dilute fine-grained syntactic signals, reducing sensitivity to hierarchical relations.

These observations suggest that SSCP's limitations are not inherent to the dual-view framework, but to the current instantiation of syntactic Bias and pooling. Future extensions could incorporate hierarchical pooling, edge-type–aware graph biases, or curriculum-based syntactic augmentation to better model complex constructions.

6.8 Summary of Findings

SCP achieves state-of-the-art performance across multiple sentence-level benchmarks. The contrastive alignment of syntax and semantics yields embeddings that are both rich in linguistic structure and discriminatively powerful. Ablations confirm that each component—

syntactic bias, graph encodings, data augmentation, and loss symmetry—contributes meaningfully.

7. CONCLUSION

In this work, we present SSCP, a contrastive learning framework that integrates syntax and semantics using a dual-encoder transformer architecture. Our results show that explicitly aligning syntactic and semantic views leads to more interpretable, robust, and generalizable sentence embeddings. Compared to prior syntax-aware models, SSCP achieves consistent gains across semantic similarity, classification, and probing benchmarks, while offering greater resilience to structural variation. The methodology exemplifies how multi-view contrastive learning can unify complementary structural and semantic information—a principle applicable to diverse domains such as code analysis, multimodal learning, and structured knowledge representation. By disentangling and aligning different perspectives of data, SSCP highlights a pathway toward richer representation learning strategies for AI systems. Future work will extend this framework to multilingual settings and to other modalities where structural and semantic dimensions interact, thereby advancing the development of more general and agentic AI models.

References

[1] Li X, Roth D.. Learning question classifiers. In Proceedings of the 19th international conference on Computational linguistics - Volume 1 (COLING '02). ACL. 2002:1–7.

[2] Xu Z, Guo D, Tang D, Su Q, Shou L, et al. Syntax-enhanced pre-trained model. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. ACL. 2021:5412-5422.

[3] Strubell E, Verga P, Andro D, McCallum A, Weiss D. Linguistically-Informed Self-Attention for Semantic Role Labelling. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP). ACL. 2018:5027-5038.

[4] Jain S, Wallace B C. Attention Is Not Explanation. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics. (NAACL-HLT). ACL. 2019:3543–3556.

[5] Wang M, Lu Z, Li H, Liu Q. Syntax-Based Deep Matching of Short Texts. Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI). AAAI Press. 2015:1354–1361.

[6] Sheng K T, Socher R, Manning D C. Improved Semantic Representations Fromtree-Structured Long Short-Term Memory Networks. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics. ACL. 2015:1556-1566.

[7] Cai R, Lapata M. Syntax-Aware Semantic Role Labelling Without Parsing. Transactions of the Association for Computational Linguistics. 2019;7:343-356.

[8] Kim Y, Jernite Y, Sontag D, Rush MA. Character-Aware Neural Language Models. Proceedings of the AAAI Conference on Artificial Intelligence. AAAI. 2016:2741-2749.

[9] Wang W, Wei F, Dong L, Bao H, Yang N, et al. StructBERT: Incorporating Language Structures Into Pre-Training for Deep Language Understanding. Proceedings of the 8th International Conference on Learning Representations. ICLR. 2020.

[10] Bai J, Wang Y, Chen Y, Yang Y, Yu J, et al. Syntax-BERT: Improving Pre-Trained Transformers With Syntax Trees. 2021. ArXiv preprint: https://arxiv.org/pdf/2103.04350

[11] Xia Q, Li Z, Zhang M. A Syntax-aware Multi-task Learning Framework for Chinese Semantic Role Labeling. 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). ACL. 2019:5382–5392.

[12] Zhang H, Zhang X, Huang H, Yu L. Prompt-Based Meta-Learning for Few-Shot Text Classification. Proceedings of the 2022 conference on empirical methods in natural language processing. ACL. 2022:1342-1357.

[13] Clark K, Khandelwal U, Levy O, D C Manning. What Does BERT Look At? An Analysis of BERT's Attention. Proceedings of ACL. ACL. 2019:276–286.

[14] Hewitt J, Manning DC. A Structural Probe for Finding Syntax in Word Representations. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. ACL. 2019;1:4129-4138.

[15] Jawahar G, Sagot B, Seddah D. What Does BERT Learn About the Structure of Language? Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. ACL. 2019:3651–3657.

[16] Reif E, Yuan A, Coenen A, Pearce A, Kim B, et al. Visualising and Measuring the Geometry Of BERT. Advances in Neural Information Processing Systems (NeurIPS). Curran Associates Inc. 2019:8592–8600.

[17] Oord DVA, Li Y, Vinyals O. Representation Learning With Contrastive Predictive Coding. 2019. arXiv preprint: https://arxiv.org/pdf/1807.03748

[18] Fang Z, Wang S, Zhou M, Ding J, Xie P. CERT: Contrastive Self-Supervised Learning for Language Understanding. 2020. ArXiv preprint: https://arxiv.org/pdf/2005.12766

[19] Wang Y, Sun C, Wu Y, Zhou H, Li L, et al. UniRE: A Unified Label Space for Entity Relation Extraction. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. ACL. 2021:220-231.

[20] Fu Y, Lin N, Yang Z, Jiang S. A Dual-Contrastive Framework for Low-Resource Cross-Lingual Named Entity Recognition. 2022. ArXiv preprint: https://arxiv.org/pdf/2204.00796

[21] Bai X, Chen Y, Song L, Zhang Y. Semantic Representation for Dialogue Modeling. 2021. ArXiv preprint: https://arxiv.org/pdf/2105.10188

[22] Devlin J, Chang M, Lee K, Toutanova K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT. ACL. 2019:4171-4186.

[23] Liu Y, Ott M, Goyal N, Du J, Joshi M, et al. RoBERTa: A Robustly Optimised Bert Pretraining Approach. 2019. ArXiv preprint: https://arxiv.org/pdf/1907.11692

[24] Clark K, Luong MT, Le QV, Manning CD. Electra: Pre-training text encoders as discriminators rather than generators. 2020. ArXiv preprint: https://arxiv.org/pdf/2003.10555

[25] Tenney I, Xia P, Chen B, Wang A, Poliak A, et al. What Do You Learn From Context? Probing for Sentence Structure in Contextualised Word Representations. International Conference on Learning Representations. ICLR. 2019.

[26] Hasan T, Bhattacharjee A, Islam SM, Mubassir K, Li FY, et al. XL-Sum: Large-Scale Multilingual Abstractive Summarisation for 44 Languages. Findings of the Association for Computational Linguistics: ACL-IJCNLP. ACL. 2021:4693–4703.

[27] Swayamdipta S, Thomson S, Lee K, Zettlemoyer L, Dyer C, et al. Syntactic Scaffolds for Semantic Structures. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. ACL. 2018:772–3782.

[28] Li B, Zhou H, He J, Wang M, Yang Y, et al. On the Sentence Embeddings From Pre-Trained Language Models. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). ACL. 2020:9119-9130.

[29] Gao T, Yao X, Chen D. SimCSE: Simple Contrastive Learning of Sentence Embeddings. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP). ACL. 2021:6894-6910.

[30] Min J, McCoy RT, Das D, Pitler E, Linzen T. Syntactic Data Augmentation Increases Robustness to Inference Heuristics. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. ACL. 2020:2339-2352.

[31] Miaschi A, Dell'Orletta F. Contextual and Non-Contextual Word Embeddings: An In-Depth Linguistic Investigation. Proceedings of the 5th Workshop on Representation Learning for NLP (RepL4NLP). ACL. 2020:110–119.

[32] Chen T, Kornblith S, Norouzi M, Hinton G. A Simple Framework for Contrastive Learning of Visual Representations. Proceedings of the 37th International Conference on Machine Learning (ICML). ACM. 2020:1597-1607.

[33] He K, Fan H, Wu Y, Xie S, Girshick R. Momentum Contrast for Unsupervised Visual Representation Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE/CVF. 2020:9729-9738.

[34] TianY, Krishnan D, Isola P. Contrastive Multiview Coding. Proceedings of the European Conference on Computer Vision. ECCV. Springer International Publishing. 2020:776–794.

[35] Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, et al. Transformers: State-Of-The-Art Natural Language Processing. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. ACL. 2020:38-45.

[36] Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, et al. Deep Contextualised Word Representations. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Proceedings of NAACL-HLT. ACL. 2018;1:2227–2237.

[37] Zhang S, Lijie W, Xiao X, Wu H. Syntax-Guided Contrastive Learning for Pre-Trained Language Model. Findings of the Association for Computational Linguistics: ACL. 2022:2430-2440.

[38] Zhang Y, Baldridge J, He H. PAWS: Paraphrase Adversaries From Word Scrambling. Proceedings of NAACL-HLT. ACL. 2019:1298-1308.

[39] https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs

[40] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. Proceedings of the 33rd International Conference on Neural Information Processing Systems. Curran Associates Inc. 2019:8026-8037.

[41] McInnes L, Healy J, Melville J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. 2020. ArXiv preprint: https://arxiv.org/pdf/1802.03426

[42] Hunter J.D. Matplotlib: A 2D Graphics Environment. Comput Sci Eng. IEEE. 2007;9:90-95.

[43] Van der Maaten L, Hinton G. Visualizing Data Using T-SNE. J Mach Learn Res. 2008;9:2579–2605.

[44] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, et al. Scikit-Learn: Machine Learning in Python. J Mach Learn Res. 2011;12:2825-2830.