Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks

INPUT: A sentence.

OUTPUT: A sentence embedding.

DATASET: SNLI + MNLI ⟶ (a total of 1 million labeled sentence pairs)

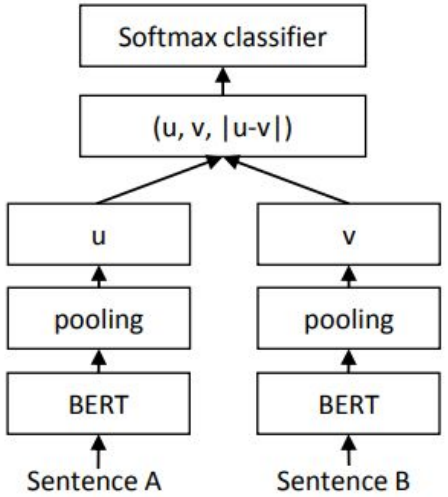


Figure 1: SBERT architecture with classification objective function, e.g., for fine-tuning on SNLI dataset. The two BERT networks have tied weights (siamese network structure).

Self-Guided Contrastive Learning for BERT Sentence Representations

INPUT: A sentence.

OUTPUT: A sentence embedding.

DATASET: Plain sentences from STS-B.

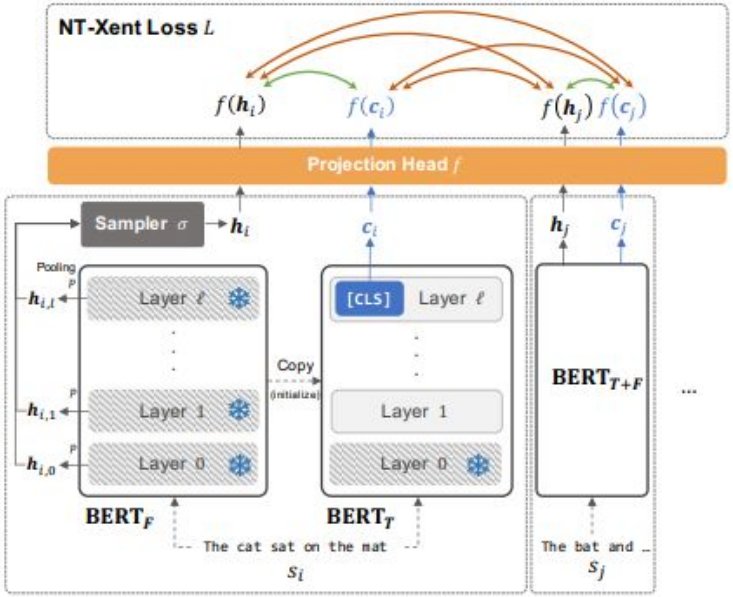


Figure 2: Self-guided contrastive learning framework. We clone BERT into two copies at the beginning of training. BERT_T (except Layer 0) is then fine-tuned to optimize the sentence vector $\mathbf{c}_i$ while BERT_F is fixed.

SimCSE: Simple Contrastive Learning of Sentence Embeddings

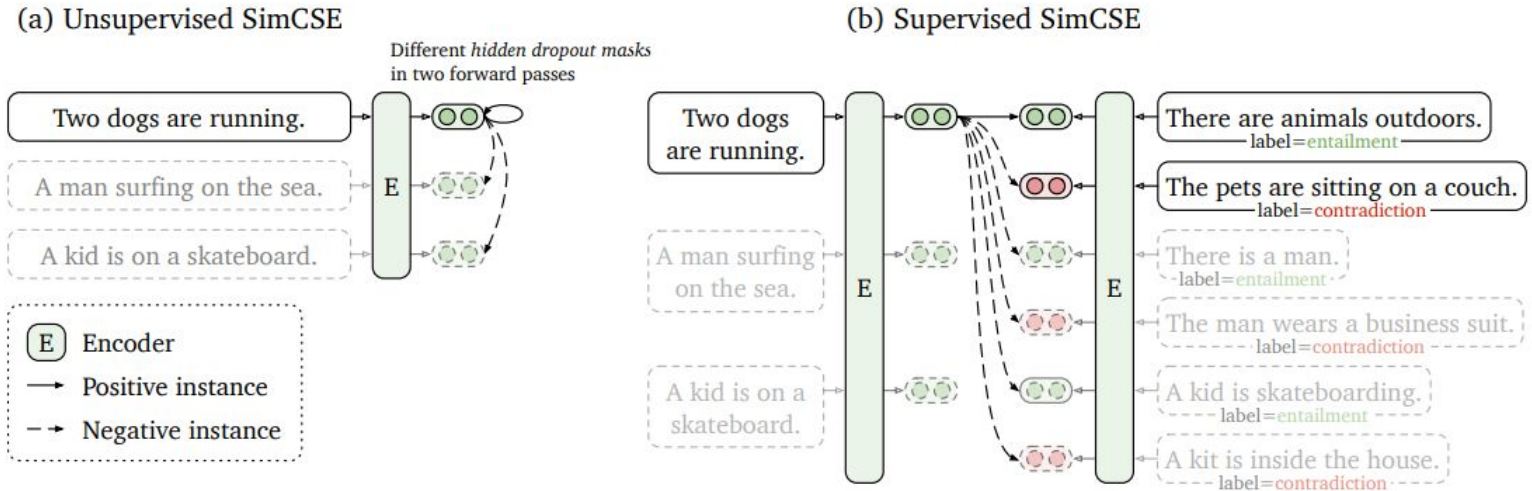
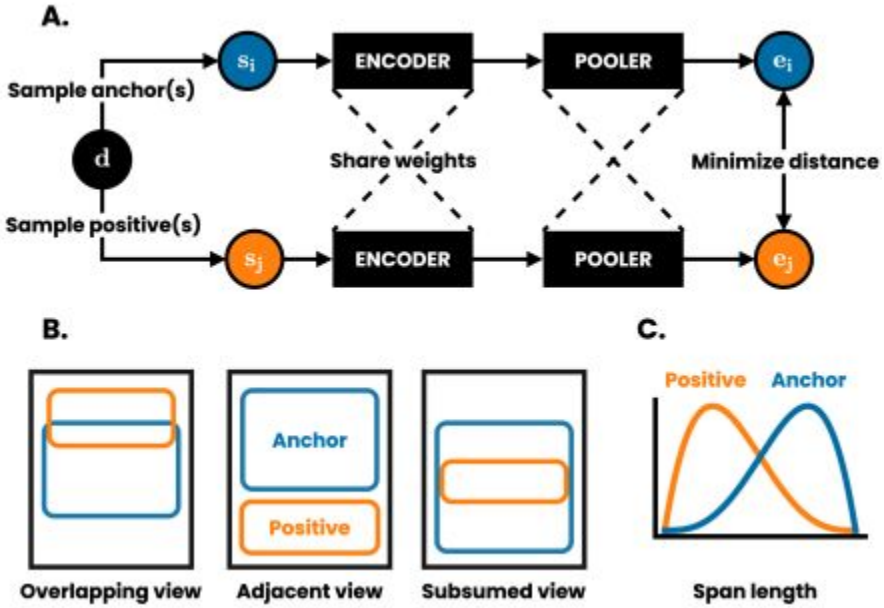


Figure 1: (a) Unsupervised SimCSE predicts the input sentence itself from in-batch negatives, with different hidden dropout masks applied. (b) Supervised SimCSE leverages the NLI datasets and takes the entailment (premise-hypothesis) pairs as positives, and contradiction pairs as well as other in-batch instances as negatives.

`INPUT:` A sentence. `OUTPUT:` Two embeddings for the same sentence with different dropout masks. `DATASET:` Randomly sampled 1 mil. sentences from English Wikipedia.

DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations



INPUT: Two spans.

OUTPUT: Two embeddings for the two spans.

DATASET: Documents with a minimum token length of 2048 from OpenWebText and WebText corpuses.

DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations

Table 5: Examples of text spans generated by our method. During training, we randomly sample one or more anchors from every document in a minibatch. For each anchor, we randomly sample one or more positives adjacent to, overlapping with, or subsumed by the anchor. All anchor-positive pairs are contrasted with every other anchor-positive pair in the minibatch. This leads to *easy* negatives (anchors and positives sampled from *other* documents in a minibatch) and *hard* negatives (anchors and positives sampled from the *same* document). Here, examples are capped at a maximum length of 64 tokens. During training, we sample spans up to a length of 512 tokens.

Anchor	Positive	Hard negative	Easy negative
		Overlapping view	
immigrant-rights advocates and law enforcement professionals were skeptical of the new program. Any effort by local cops to enforce immigration laws, they felt, would be bad for community policing, since immigrant victims or witnesses of crime wouldn't feel comfortable talking to police.	feel comfortable talking to police. Some were skeptical that ICE's intentions were really to protect public safety, rather than simply to deport unauthorized immigrants more easily.	liberal parts of the country with large immigrant populations, like Santa Clara County in California and Cook County in Illinois, agreed with the critics of Secure Communities. They worried that implementing the program would strain their relationships with immigrant residents.	that a new location is now available for exploration. A good area, in my view, feels like a natural progression of a game world it doesn't seem tacked on or arbitrary. That in turn needs it to relate
		Adjacent view	
if the ash stops belching out of the volcano then, after a few days, the problem will have cleared, so that's one of the factors. "The other is the wind speed and direction." At the moment the weather patterns are very volatile which is what is making it quite difficult, unlike last year, to predict	where the ash will go. "The public can be absolutely confident that airlines are only able to operate when it is safe to do so." Ryanair said it could not see any ash cloud	A British Airways jumbo jet was grounded in Canada on Sunday following fears the engines had been contaminated with volcanic ash	events are processed in FIFO order. When this nextTickQueue is emptied, the event loop considers all operations to have been completed for the current phase and transitions to the next phase.
		Subsumed view	
Far Cry Primal is an action-adventure video game developed by Ubisoft Montreal and published by Ubisoft. It was released worldwide for PlayStation 4 and Xbox One on February 23, 2016, and for Microsoft Windows on March 1, 2016. The game is a spin-off of the main Far Cry series. It is the first Far Cry game set in the Mesolithic Age.	by Ubisoft. It was released worldwide for PlayStation 4 and Xbox One on February 23, 2016, and for Microsoft Windows on March 1, 2016. The game is a spin-off of the main Far Cry series.	Players take on the role of a Wenja tribesman named Takkar, who is stranded in Oros with no weapons after his hunting party is ambushed by a Saber-tooth Tiger.	to such feelings. Fawkes cried out and flew ahead, and Albus Dumbledore followed. Further along the Dementors' path, people were still alive to be fought for. And no matter how much he himself was hurting, while there were still people who needed him he would go on. For

An Unsupervised Sentence Embedding Method by Mutual Information Maximization

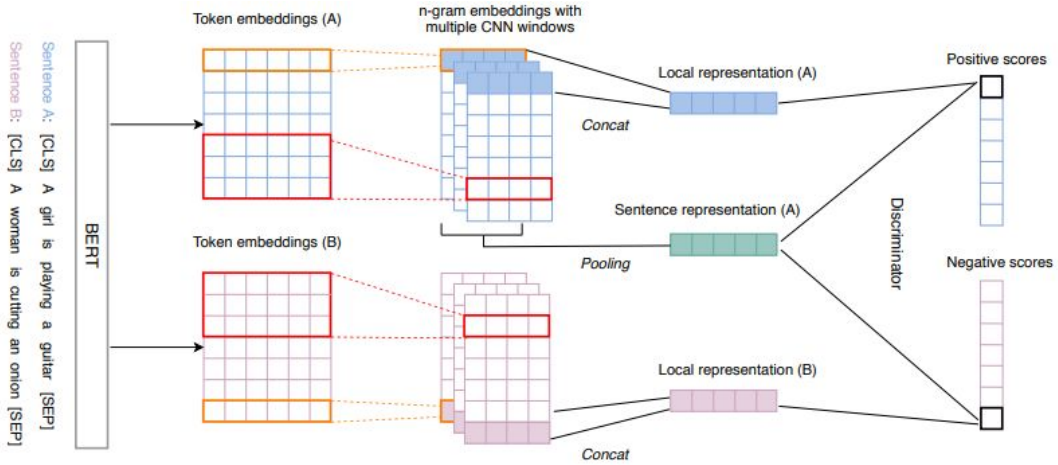


Figure 1: Model Architecture. Two sentences are encoded by BERT and multiple CNNs with different window sizes to get concatenated local n-gram token embeddings. A discriminator T takes all pairs of {sentence representation, token representation} as input and decides whether they are from the same sentence. In this example, we treat sentence "A" as the positive sample and "B" as negative, then n-gram embeddings of "A" will be summarized to a global sentence embedding via pooling. The discriminator produces scores for all token representations from both "A" and "B" to maximize the MI estimator in Eq.2.

`INPUT:` Two sentences.

`OUTPUT:` Token scores produced by the discriminator.

`DATASET:` IS-BERT-NLI (SNLI + MNLI without labels)

Deep Continuous Prompt for Contrastive Learning of Sentence Embeddings

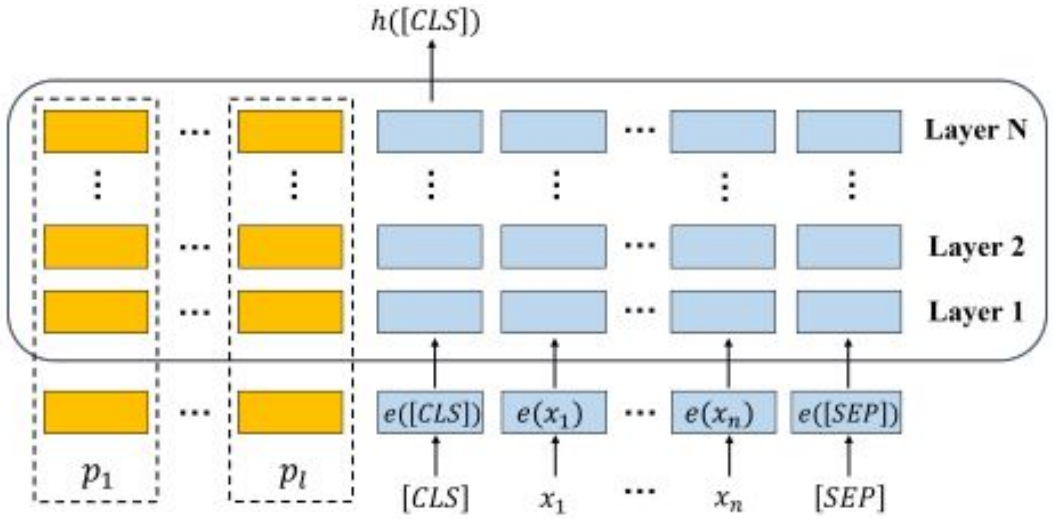


Figure 1: Deep continuous prompt framework for contrastive learning of sentence embeddings. We freeze the transformer parameters (the blue blocks) and only optimize the prefix deep continuous prompts (the orange blocks).

INPUT: A sentence.

OUTPUT: An embeddings for the sentence.

DATASET: Randomly sampled 1 mil. sentences from English Wikipedia.

DiffCSE: Difference-based Contrastive Learning for Sentence Embeddings

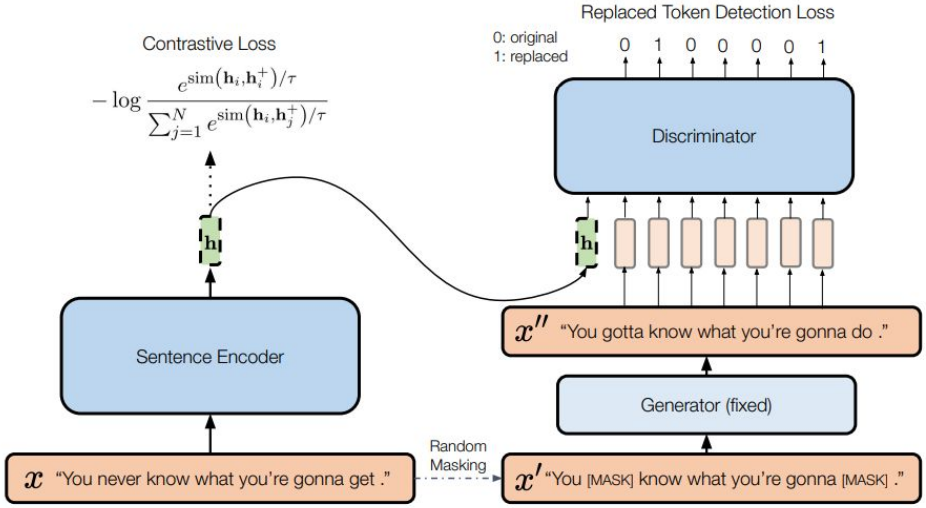


Figure 1: Illustration of DiffCSE. On the left-hand side is a standard SimCSE model trained with regular contrastive loss on dropout transformations. On the right hand side is a conditional difference prediction model which takes the sentence vector $\mathbf{h}$ as input and predict the difference between x and x''. During testing we discard the discriminator and only use $\mathbf{h}$ as the sentence embedding.

INPUT: A sentence.

OUTPUT: Replaced token predictions.

DATASET: Randomly sampled 1 mil. sentences from English Wikipedia.

ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer

INPUT: A sentence.

OUTPUT: A sentence embedding.

DATASET: Unlabeled versions of 7 datasets in the SentEval tool.

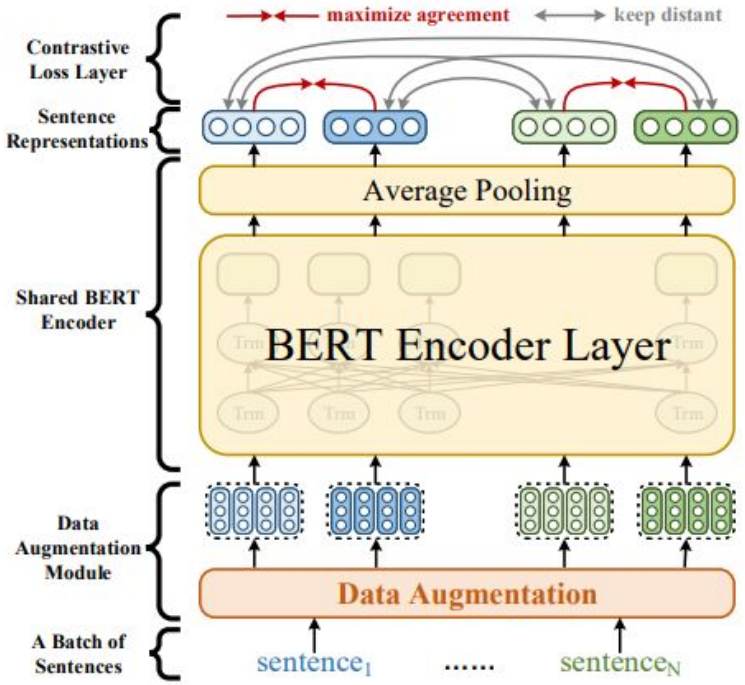


Figure 2: The general framework of our proposed approach.

ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer

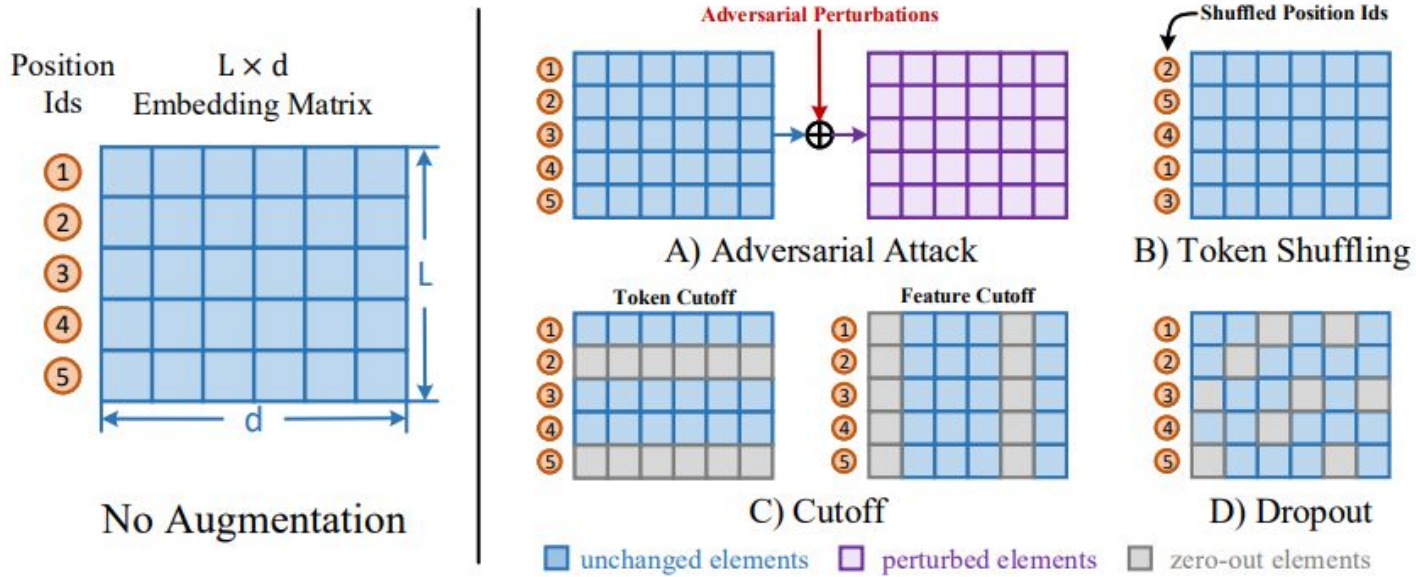


Figure 3: The four data augmentation strategies used in our experiments.

Adversarial Attack ⟶ adds a worst-case perturbation to the input sample.

Token Shuffling ⟶ randomly shuffles the order of tokens in the input sequence.

Cutoff ⟶ randomly erases some tokens (for token cutoff), feature dimensions (for feature cutoff), or token spans (for span cutoff).

Dropout ⟶ randomly drops elements in the token embedding layer and sets their values to zero

CLEAR: Contrastive Learning for Sentence Representation

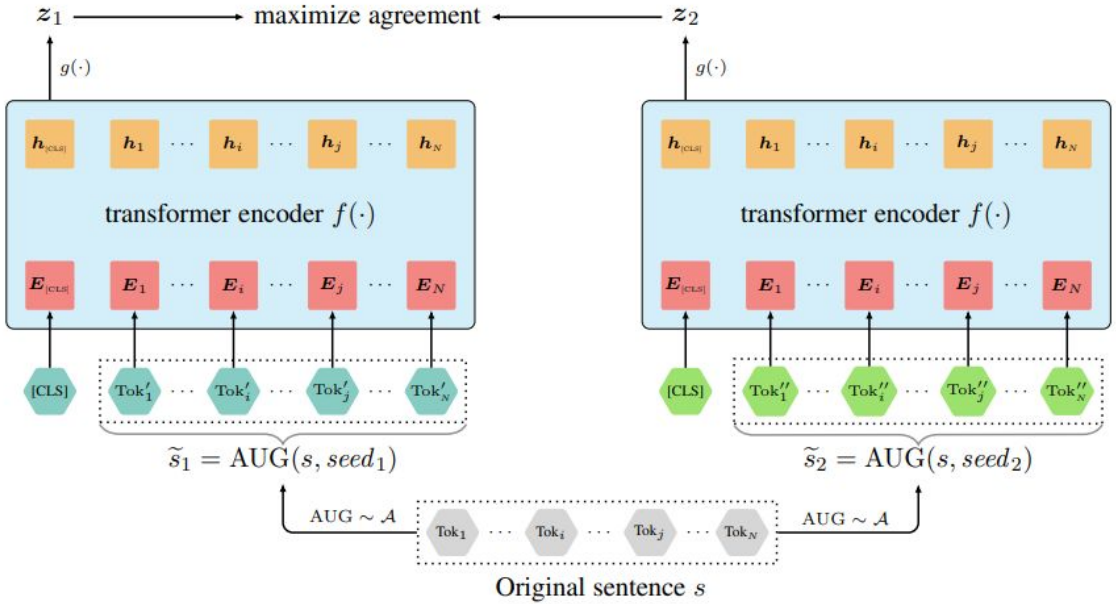
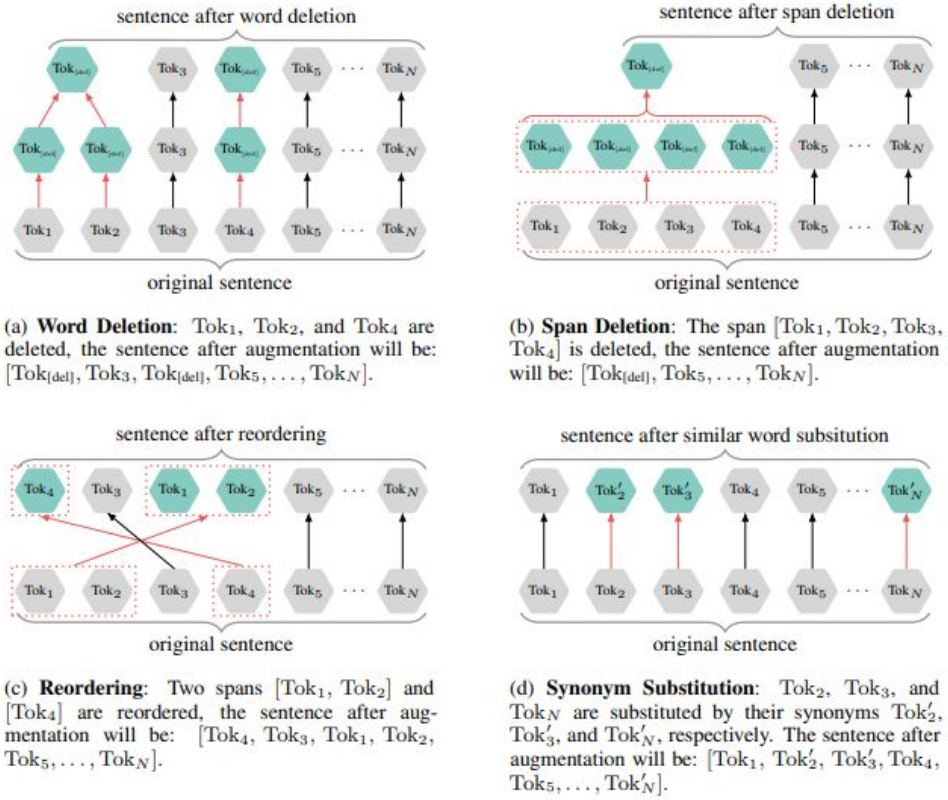


Figure 1: The proposed contrastive learning framework CLEAR.

INPUT: A sentence. OUTPUT: A sentence embedding.

DATASET: Combination of BookCorpus and English Wikipedia.

CLEAR: Contrastive Learning for Sentence Representation



(a) **Word Deletion**: Tok_1, Tok_2, and Tok_4 are deleted, the sentence after augmentation will be: $[\text{Tok}_{[\text{del}]}, \text{Tok}_3, \text{Tok}_{[\text{del}]}, \text{Tok}_5, \ldots, \text{Tok}_N]$.

(b) **Span Deletion**: The span $[\text{Tok}_1, \text{Tok}_2, \text{Tok}_3, \text{Tok}_4]$ is deleted, the sentence after augmentation will be: $[\text{Tok}_{[\text{del}]}, \text{Tok}_5, \ldots, \text{Tok}_N]$.

(c) **Reordering**: Two spans $[\text{Tok}_1, \text{Tok}_2]$ and $[\text{Tok}_4]$ are reordered, the sentence after augmentation will be: $[\text{Tok}_4, \text{Tok}_3, \text{Tok}_1, \text{Tok}_2, \text{Tok}_5, \ldots, \text{Tok}_N]$.

(d) **Synonym Substitution**: Tok_2, Tok_3, and Tok_N are substituted by their synonyms Tok'_2, Tok'_3, and Tok'_N, respectively. The sentence after augmentation will be: $[\text{Tok}_1, \text{Tok}'_2, \text{Tok}'_3, \text{Tok}_4, \text{Tok}_5, \ldots, \text{Tok}'_N]$.

Figure 2: Four sentence augmentation methods in proposed contrastive learning framework CLEAR.

Word deletion ⟶ randomly selects tokens in the sentence and replace them with a special token [DEL].

Span deletion ⟶ picks and replaces the deletion objective on the span level. Generally, a special case of word-deletion, which focuses deleting consecutive words.

Reordering ⟶ randomly samples several pairs of span and switch them pairwise to construct the reordered versions.

Synonym substitution ⟶ samples some words and replaces them with synonyms.

COCO-LM: Correcting and Contrasting Text Sequences for Language Model Pretraining

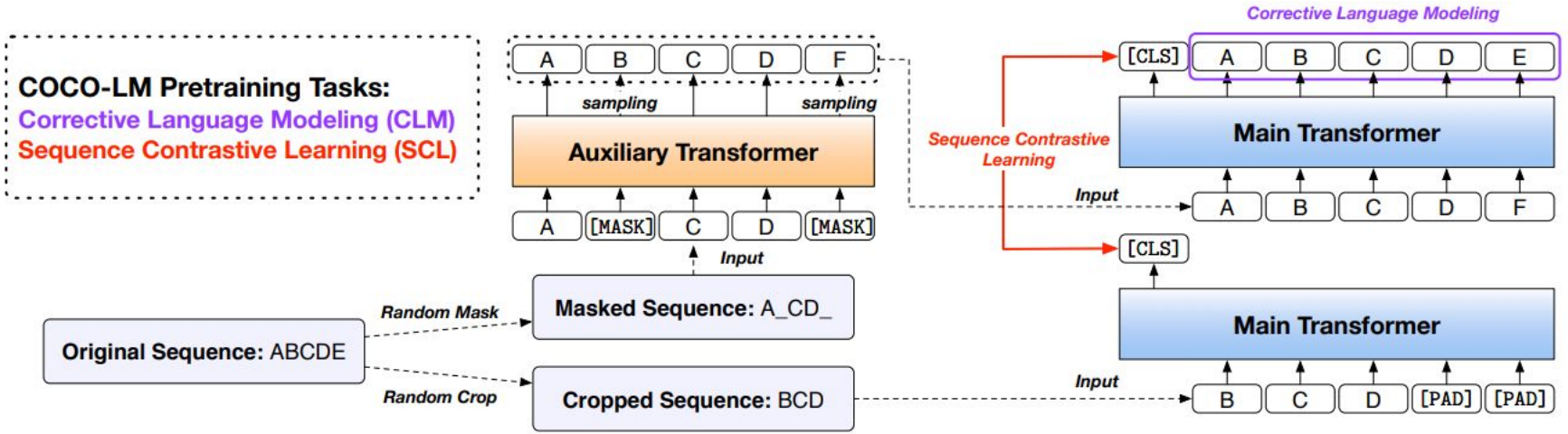
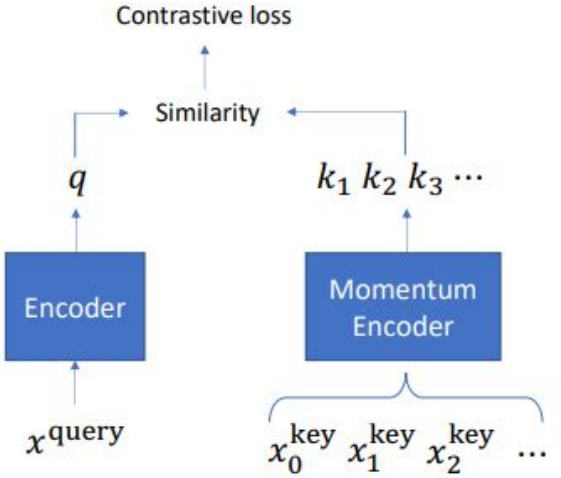


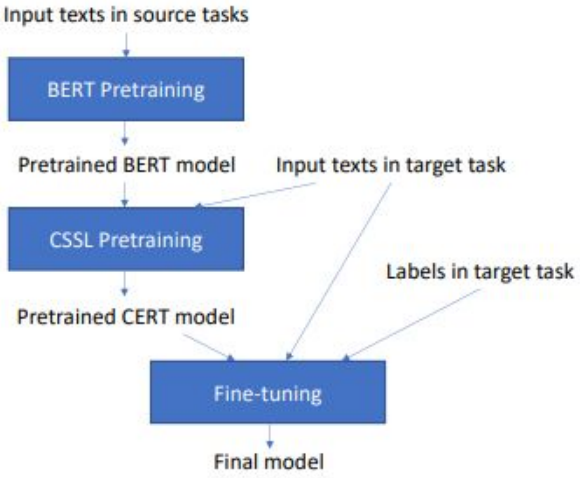
Figure 2: The overview of COCO-LM. The auxiliary Transformer is pretrained by MLM. Its corrupted text sequence is used as the main Transformer's pretraining input in Corrective Language Modeling and paired with the cropped original sequence for Sequence Contrastive Learning.

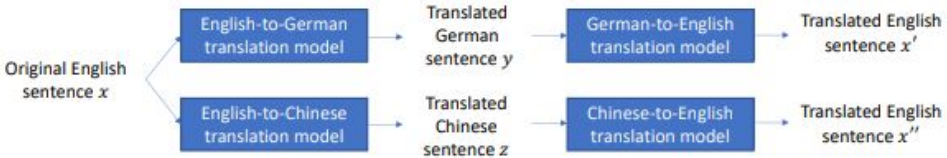
INPUT: A sentence. OUTPUT: Two sentence embeddings.

DATASET: Combination of BookCorpus and English Wikipedia.

CERT: Contrastive Self-supervised Learning for Language Understanding







INPUT: A sentence.

OUTPUT: Two sentence embeddings.

DATASET: Data from the datasets of target tasks (i.e. CoLA, STS-B).

PromptBERT: Improving BERT Sentence Embeddings with Prompts

INPUT: A sentence.

OUTPUT: A sentence embedding.

DATASET: A combination of English Wikipedia and NLI datasets.

Template	STS-B dev.
Searching for relationship tokens	
[X] [MASK] .	39.34
[X] is [MASK] .	47.26
[X] mean [MASK] .	53.94
[X] means [MASK] .	63.56
Searching for prefix tokens	
This [X] means [MASK] .	64.19
This sentence of [X] means [MASK] .	68.97
This sentence of "[X]" means [MASK] .	70.19
This sentence : "[X]" means [MASK] .	73.44

Table 4: Greedy searching templates on *bert-base-uncased*.

Universal Sentence Representation Learning with Conditional Masked Language Model

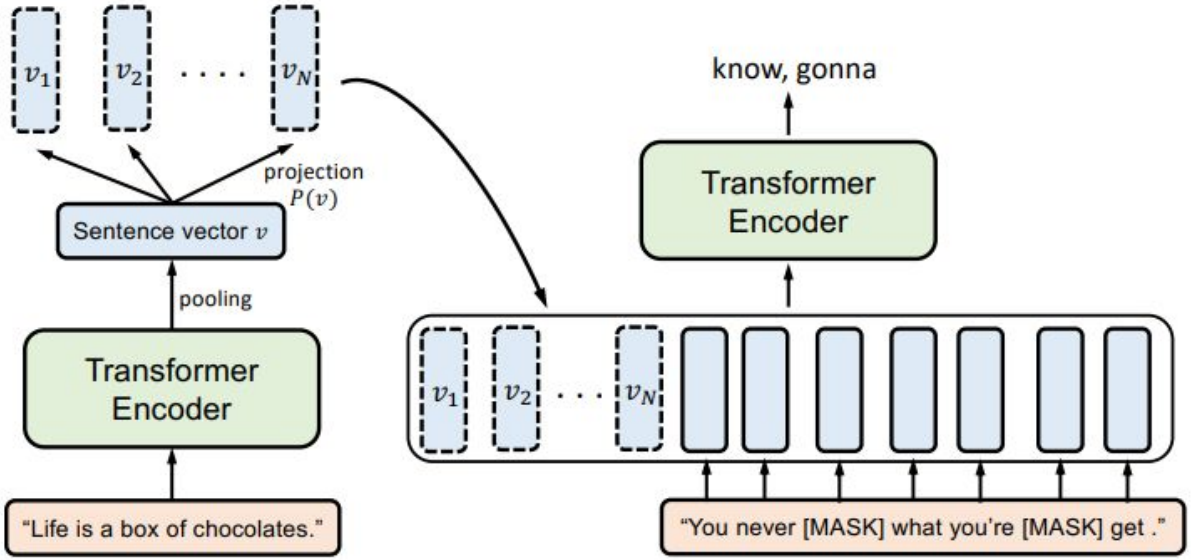


Figure 1: The architecture of Conditional Masked Language Modeling (CMLM).

INPUT: A sentence. OUTPUT: Two sentence embeddings. DATASET: Three Common Crawl dumps.

	STS12	STS13	STS14	STS15	STS16	STSb	SICK-R	Avg.
SBERT-NLI-large	72.27	78.46	74.90	80.99	76.25	79.23	73.75	76.55
SBERT-large+ SG-OPT	**74.58**	83.79	76.98	84.57	79.87	**82.05**	76.44	**79.76**
SimCSE RoBERTa-large	72.86	83.99	75.62	84.77	**81.80**	81.98	71.26	78.90
DeCLUTR-base	63.56	72.58	71.70	79.95	79.59	79.39	**78.62**	75.05
IS-BERT-NLI	56.77	69.24	61.21	75.23	70.16	69.21	64.25	66.58
SimCSE-BERT DCPCSE-BERT	73.34	**85.90**	77.10	**85.26**	80.08	80.96	73.28	79.42
DiffCSE-BERTbase	72.28	84.43	76.47	83.90	80.54	80.59	71.23	78.49
ConSERT-large	70.69	82.96	74.13	82.78	76.66	77.53	70.37	76.45
CLEAR-MLM2	49.0	48.9	57.4	63.6	65.6	72.5	75.6	61.8
CMLM-base	58.20	61.07	61.67	73.32	74.88	76.60	64.80	67.22
PromptRoBERTa-base	73.94	84.74	**77.28**	84.99	81.74	81.88	69.50	79.15