# Contrastive Representation Learning in Text

Danqi Chen

@danqi_chen @princeton_nlp

November 18, 2021

# Contrastive learning

Learning representations by contrasting **positive** and **negative** examples
(Hadsell et al., 2006)

(Image credit: Ekin Tiu)

$$\mathrm{sim}(f(x), f(x^+)) \gg \mathrm{sim}(f(x), f(x^-))$$

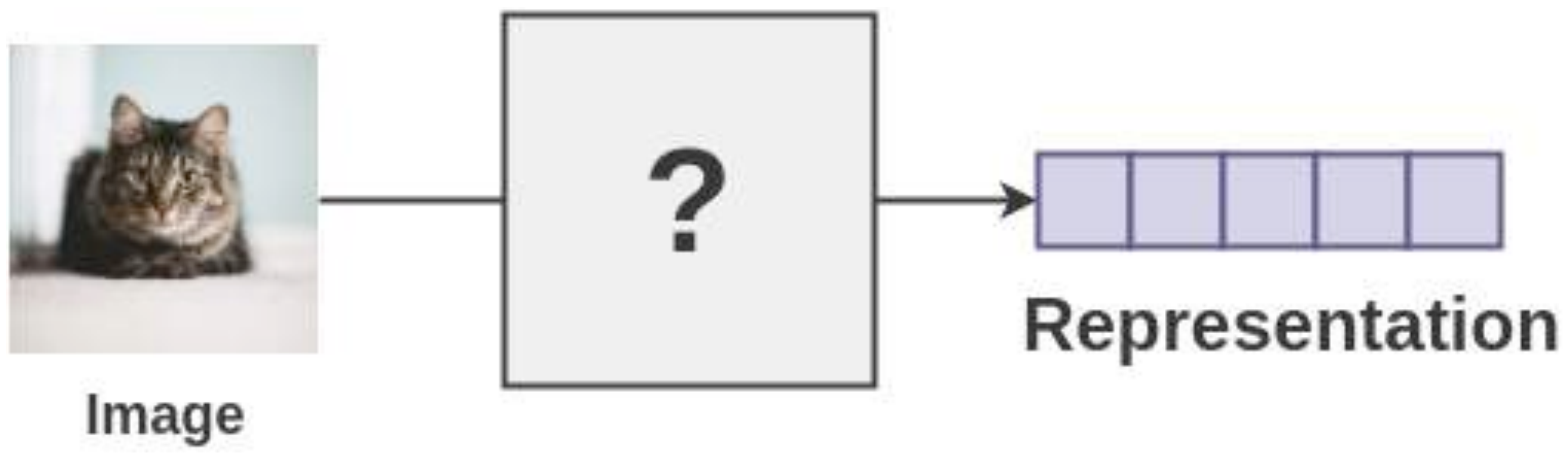
***f*: encoder**

$x$: anchor
$x^+$: positive example,
$x^-$: negative example

# Contrastive learning of visual representations

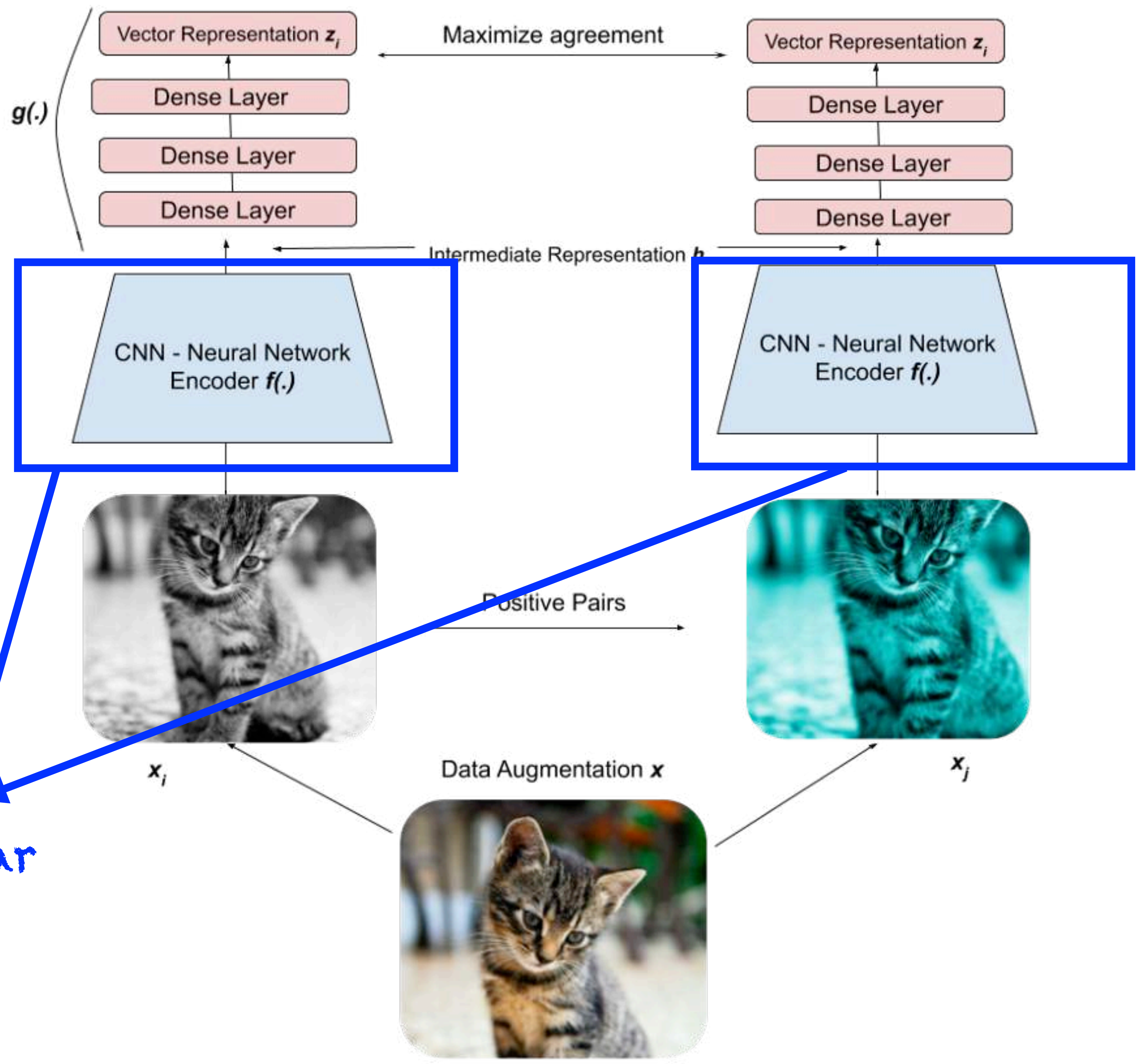
SimCLR (Chen et al., 2020), MoCo (He et al., 2020), SwAV (Caron et al., 2020) and many others

***f*: encoder**




- **positive pairs** = two random transformations of the **same image**
- **negative pairs** = the transformations of **other images** in the same mini-batch

CNN encoder: training a linear classifier or fine-tuning




SimCLR (Chen et al., 2020)

# Contrastive learning of visual representations

SimCLR (Chen et al., 2020), MoCo (He et al., 2020), SwAV (Caron et al., 2020) and many others

**InfoNCE loss**

$$\mathcal{L}_N = -\mathbb{E}_X \left( \log \frac{\exp(\text{sim}(f(x), f(x^+)))}{\exp(\text{sim}(f(x), f(x^+))) + \sum_{j=1}^{N-1} \exp(\text{sim}(f(x), f(x_j)))} \right)$$

1 positive example +
*N*-1 negative examples (in-batch negatives)

**Key ingredients:**

- Where do **positive pairs** come from (e.g., **data augmentation**)?
- The impact of batch size (= how many **negatives**)?
- Hard negatives

# What is the analogy in text?

- Most successful example: word2vec (Mikolov et al., 2013)

Two encoders instead of one!

**positive pairs** = (center word, **context** word)

**negative pairs** = (center word, **random** word)

**positive examples +**

| $w$ | $c_{\text{pos}}$ |
|---|---|
| apricot | tablespoon |
| apricot | of |
| apricot | jam |
| apricot | a |

**negative examples -**

| $w$ | $c_{\text{neg}}$ | $w$ | $c_{\text{neg}}$ |
|---|---|---|---|
| apricot | aardvark | apricot | seven |
| apricot | my | apricot | forever |
| apricot | where | apricot | dear |
| apricot | coaxial | apricot | if |

(Image credit: SLP3)

# What is the analogy in the BERT era?

store ↑ gallon ↑

the man went to the [MASK] to buy a [MASK] of milk

- RQ1. **When** and **why** does contrastive learning work with **pre-trained language representations**?
- RQ2. Why not contrastive learning in **pre-training**?

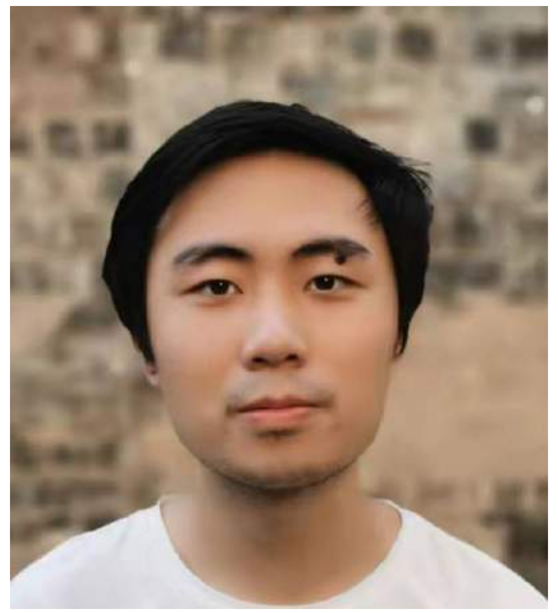
# This talk

- **Learning universal sentence representations**
  - SimCSE (Gao et al., EMNLP 2021)
- **Learning dense representations for retrieval**
  - DPR (Karpukhin et al., EMNLP 2020)
  - DensePhrases (Lee et al., ACL 2021; Lee et al., EMNLP 2021)

- RQ1. When and why does contrastive learning work with **pre-trained language representations**?
- RQ2. Why not contrastive learning in pre-training?

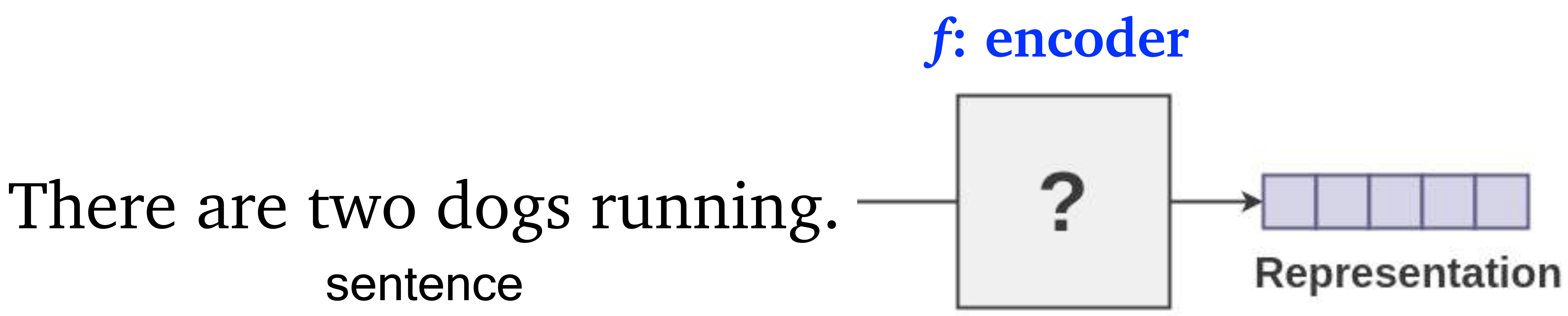
# SimCSE: Simple Contrastive Learning of Sentence Embeddings

(Work done by Tianyu Gao and Xingcheng Yao)



https://github.com/princeton-nlp/SimCSE

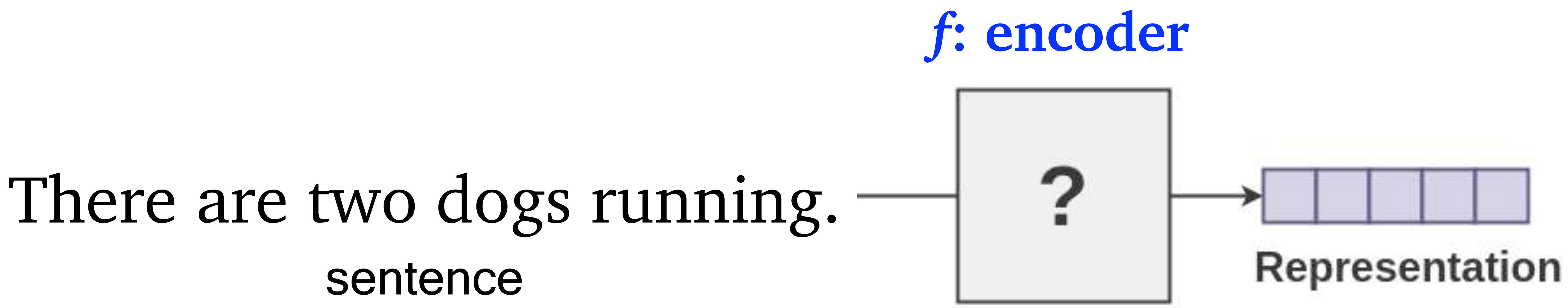
# Universal sentence embeddings




**Applications:**

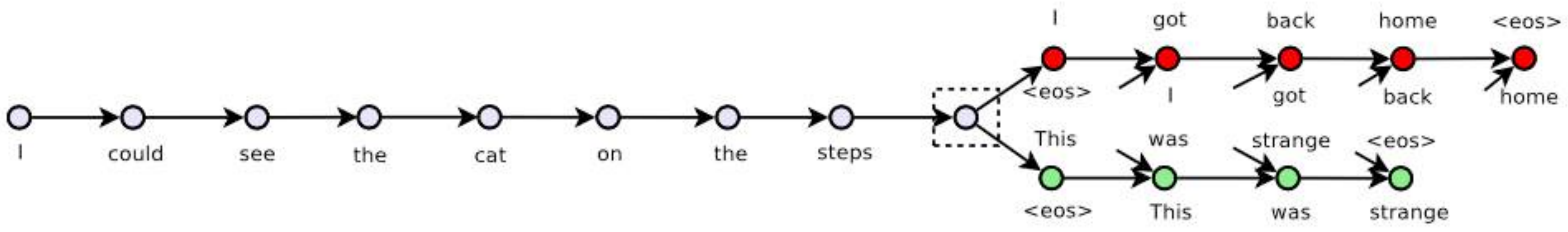
- **Clustering** (e.g., topic modeling)
- **Retrieval** (e.g., semantic search)
- **Transfer learning** to other NLP tasks
  (e.g., training a linear classifier for sentiment analysis)

# Universal sentence embeddings

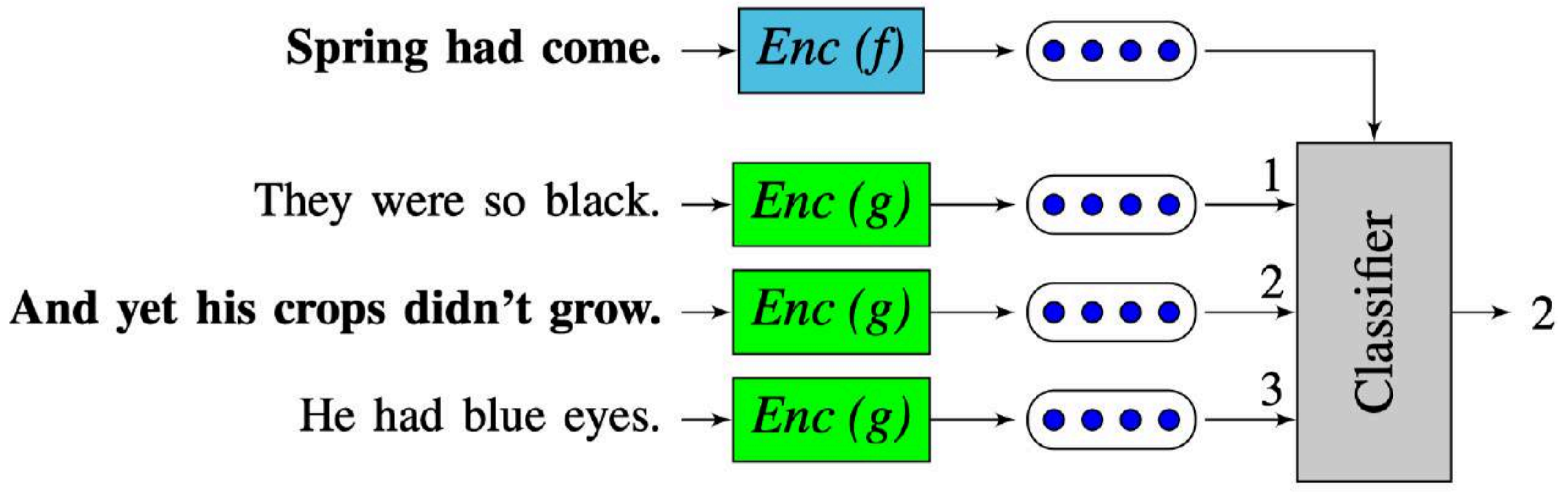



- **Previous work:** use the current sentence to **predict next or previous sentence**

Skip-thought (Kiros et al., 2015)



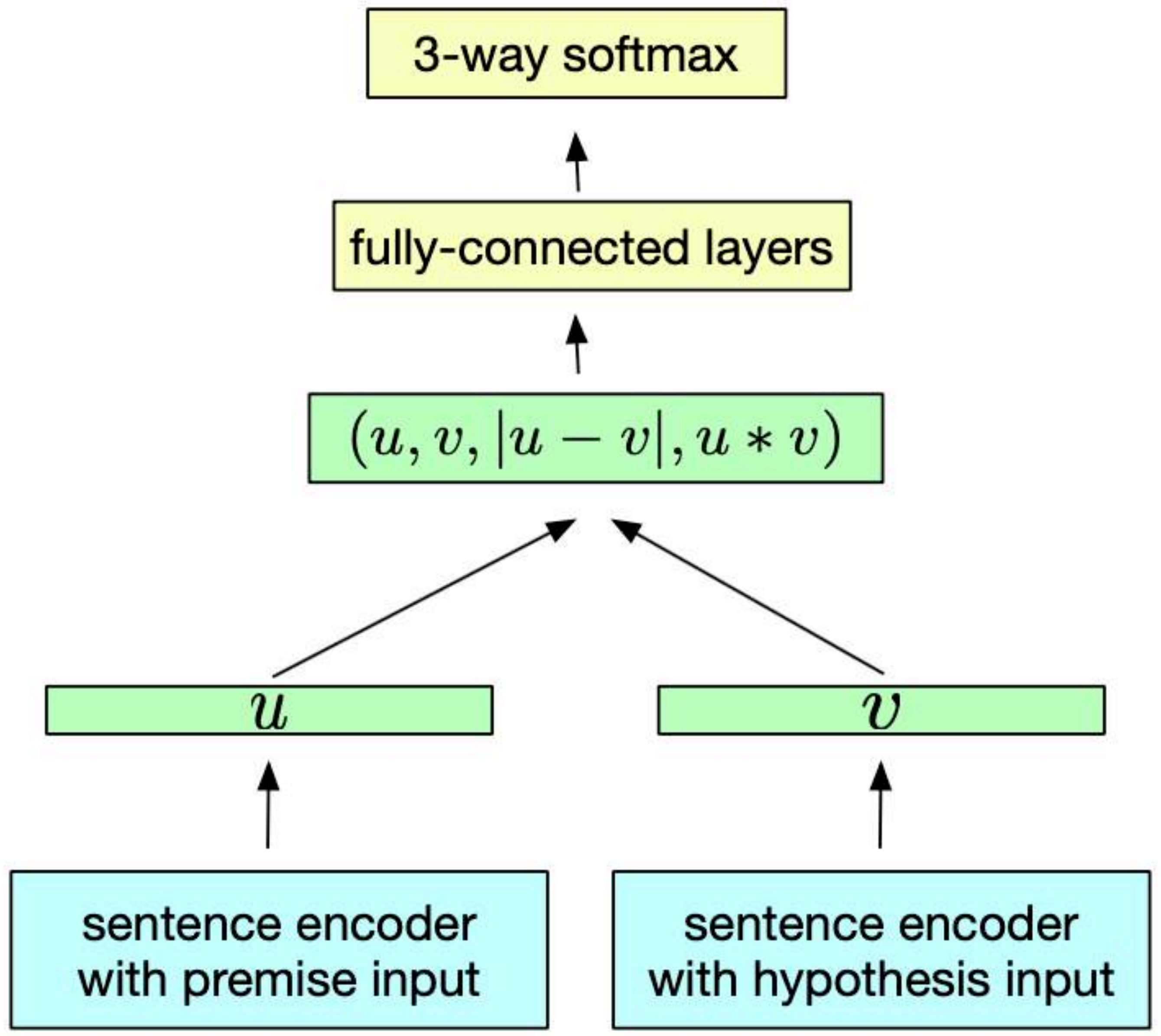

QuickThoughts (Logeswaran et al., 2018)




Contrastive learning

# Universal sentence embeddings

- **Previous work:** learning from **natural language inference (NLI) datasets**




**Natural language inference**

premise = A soccer game with multiple males playing.

hypothesis = Some men are playing a sport.

label = {**entailment**, contradiction, neutral}

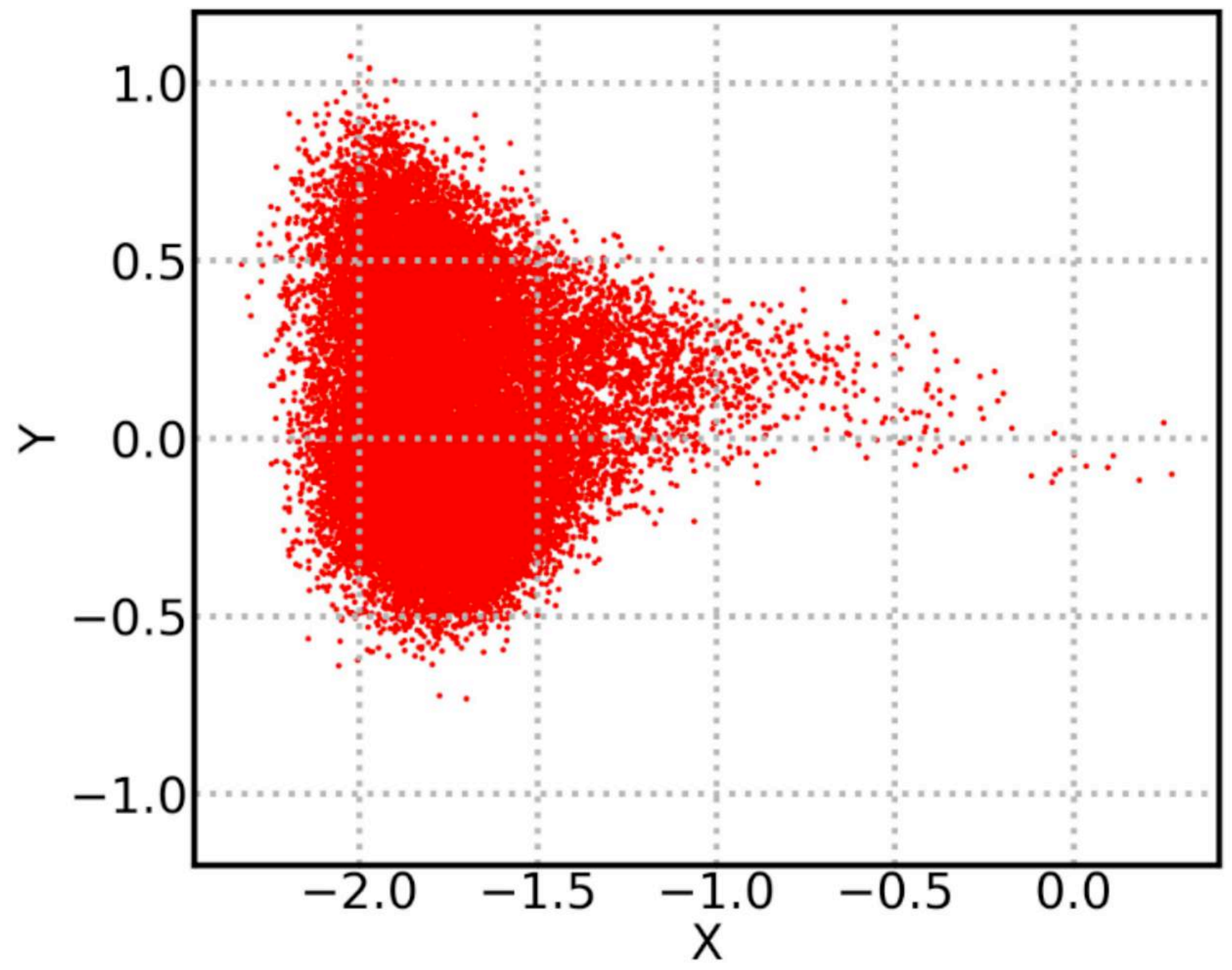
InferSent (Conneau et al., 2017): LSTMs

SentenceBERT (Reimers and Gurevych, 2019): BERT

# Universal sentence embeddings

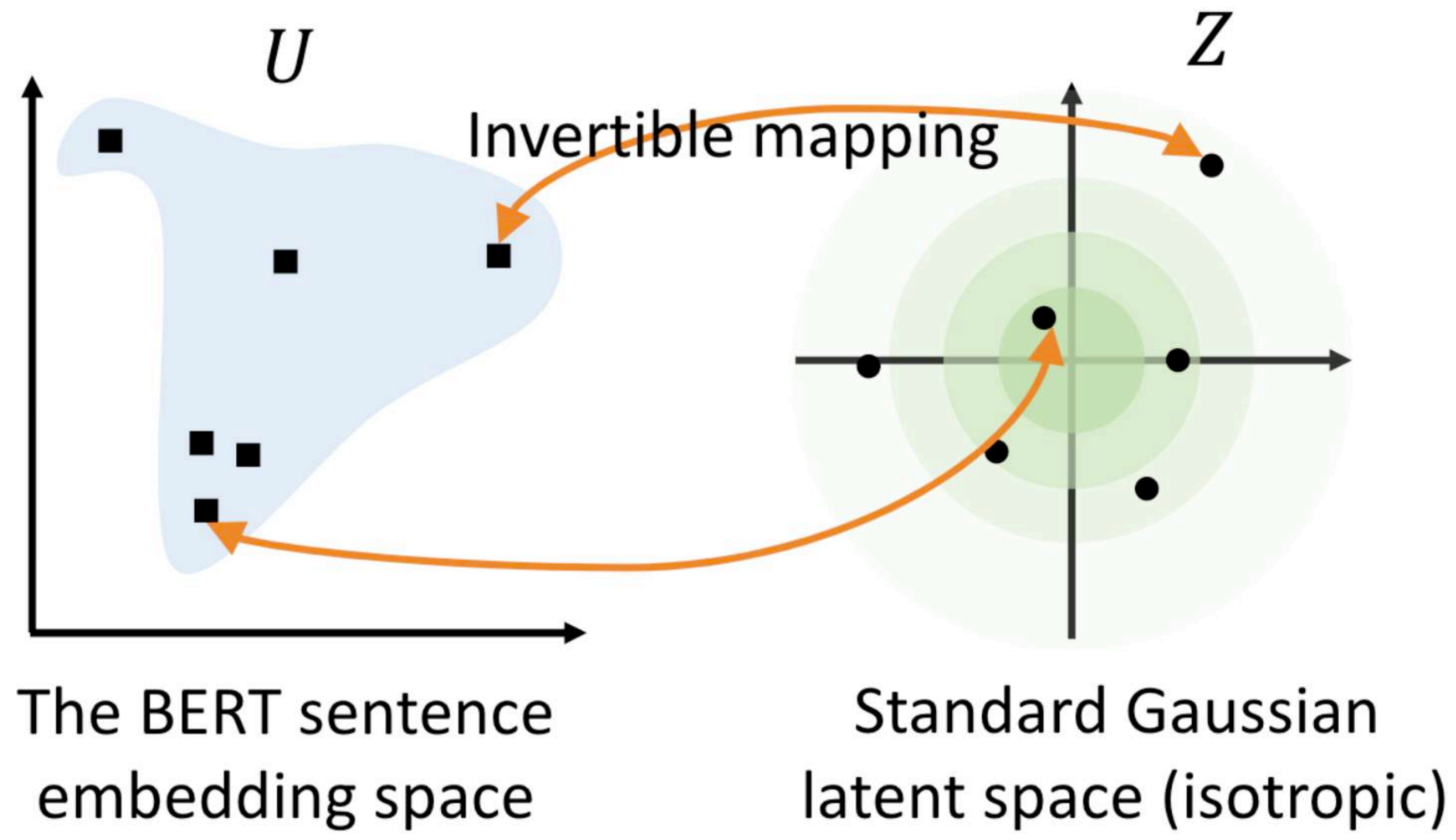
Q: Why can't we directly obtain sentence embeddings from BERT (e.g., average)?

**Issue**: pre-trained embeddings are **highly anisotropic** (Gao et al., 2019; Ethayarajh, 2019; Li et al., 2020)




(Gao et al., 2019)

**Solution**: post-processing and mapping embeddings to an isotropic space



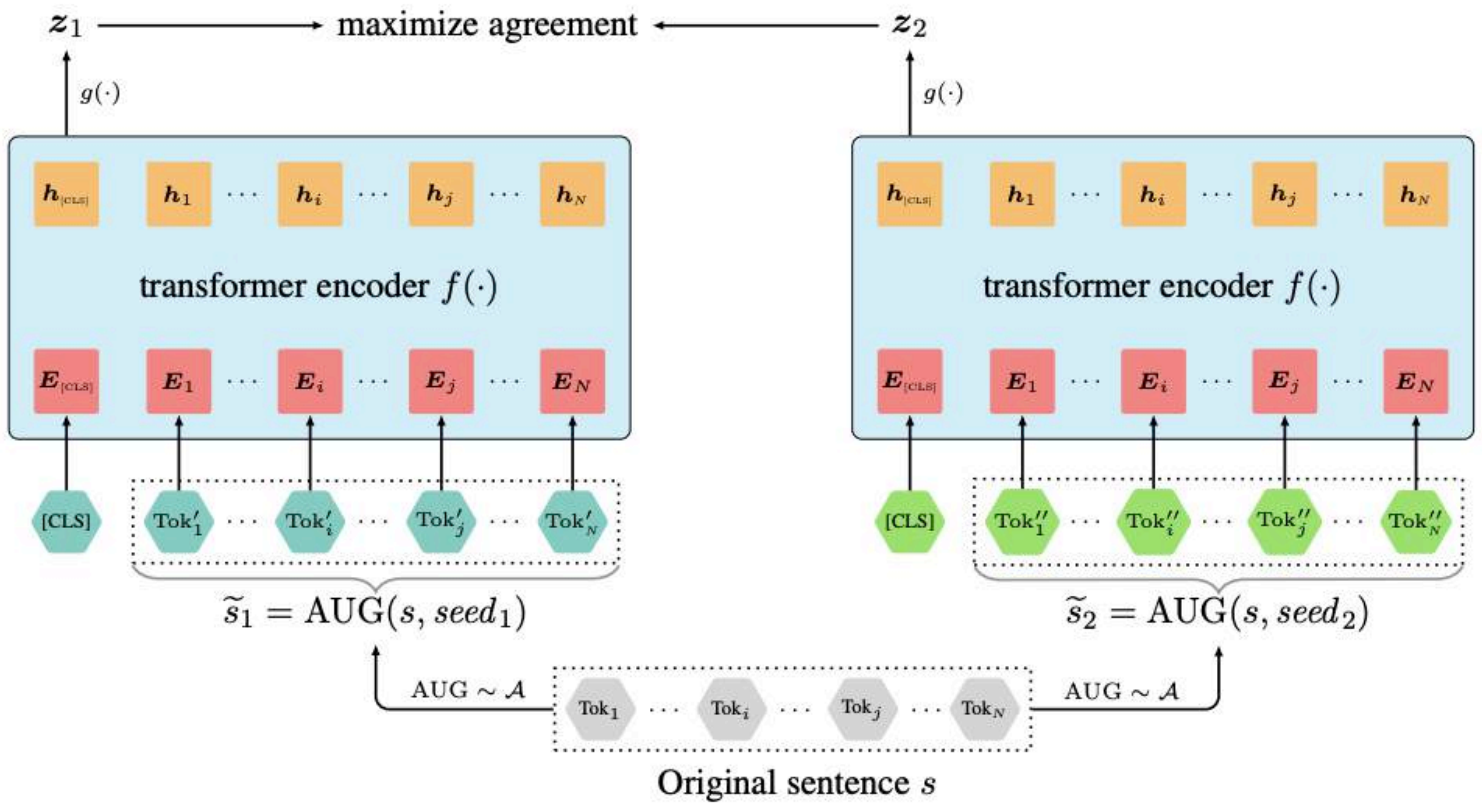

BERT-flow (Li et al., 2020)

BERT-whitening (Su et al., 2021)

# Universal sentence embeddings

Q: Can we apply the SimCLR idea to sentence representations?

**CLEAR** (Wu et al., 2020)




**Data augmentation**: word/span deletion, reordering, synonym substitution

The performance is not competitive. **Why?**

# Our approach: SimCSE

A simple contrastive learning framework for sentence representations:

- Unsupervised SimCSE: only uses ***dropout*** as data augmentation
- Supervised SimCSE: uses **entailment** + **contradiction** pairs from NLI datasets

**InfoNCE loss**

$$\mathcal{L}_N = -\mathbb{E}_X \left( \log \frac{\exp(\mathrm{sim}(f(x), f(x^+)))}{\exp(\mathrm{sim}(f(x), f(x^+))) + \sum_{j=1}^{N-1} \exp(\mathrm{sim}(f(x), f(x_j)))} \right)$$

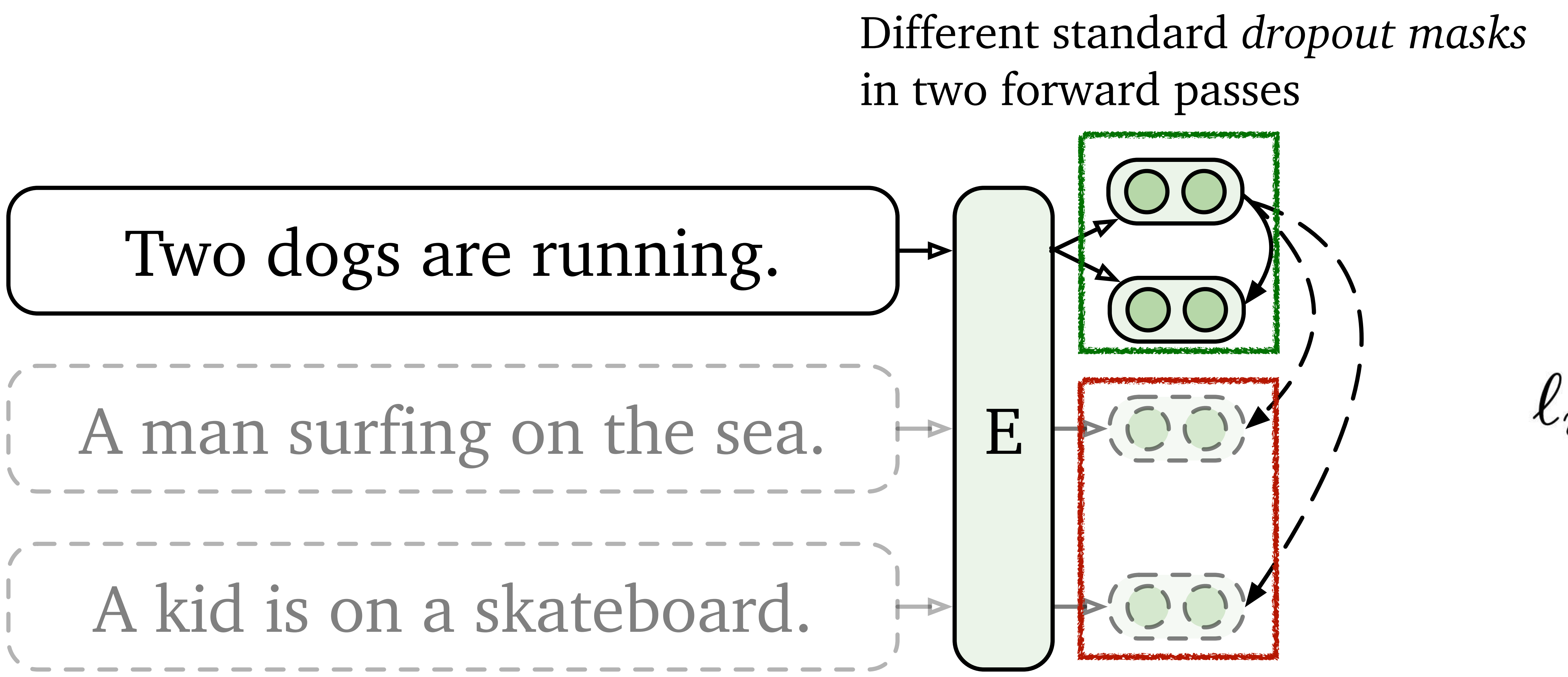
$x$: a sentence, $f(\cdot)$: BERT encoder "[CLS]" + fine-tuning

Key: how to find positive and negative pairs?

# Unsupervised SimCSE

**Positive pairs:** embeddings of the **same sentence** with **different dropout masks**

**Negative pairs:** embeddings of other sentences (in-batch negatives)




Same sentence with different dropout

$$\ell_i = -\log \frac{e^{\mathrm{sim}(\mathbf{h}_i^{z_i}, \mathbf{h}_i^{z_i'})/\tau}}{\sum_{j=1}^{N} e^{\mathrm{sim}(\mathbf{h}_i^{z_i}, \mathbf{h}_j^{z_j'})/\tau}}$$

In-batch negatives

sim: cosine similarity

$\tau$: temperature term

# Supervised SimCSE

**Positive pairs: entailment** (premise, hypothesis) pairs

**Negative pairs: contradiction** (premise, hypothesis) pairs + in-batch negatives

Given one premise,

- Premise: *There are two dogs running.*
- **Entailment: *There are animals outdoors.***
- **Contradiction: *The pets are sitting on a couch.***
- ~~Neutral: *The dogs are catching a ball.*~~

Positive pairs

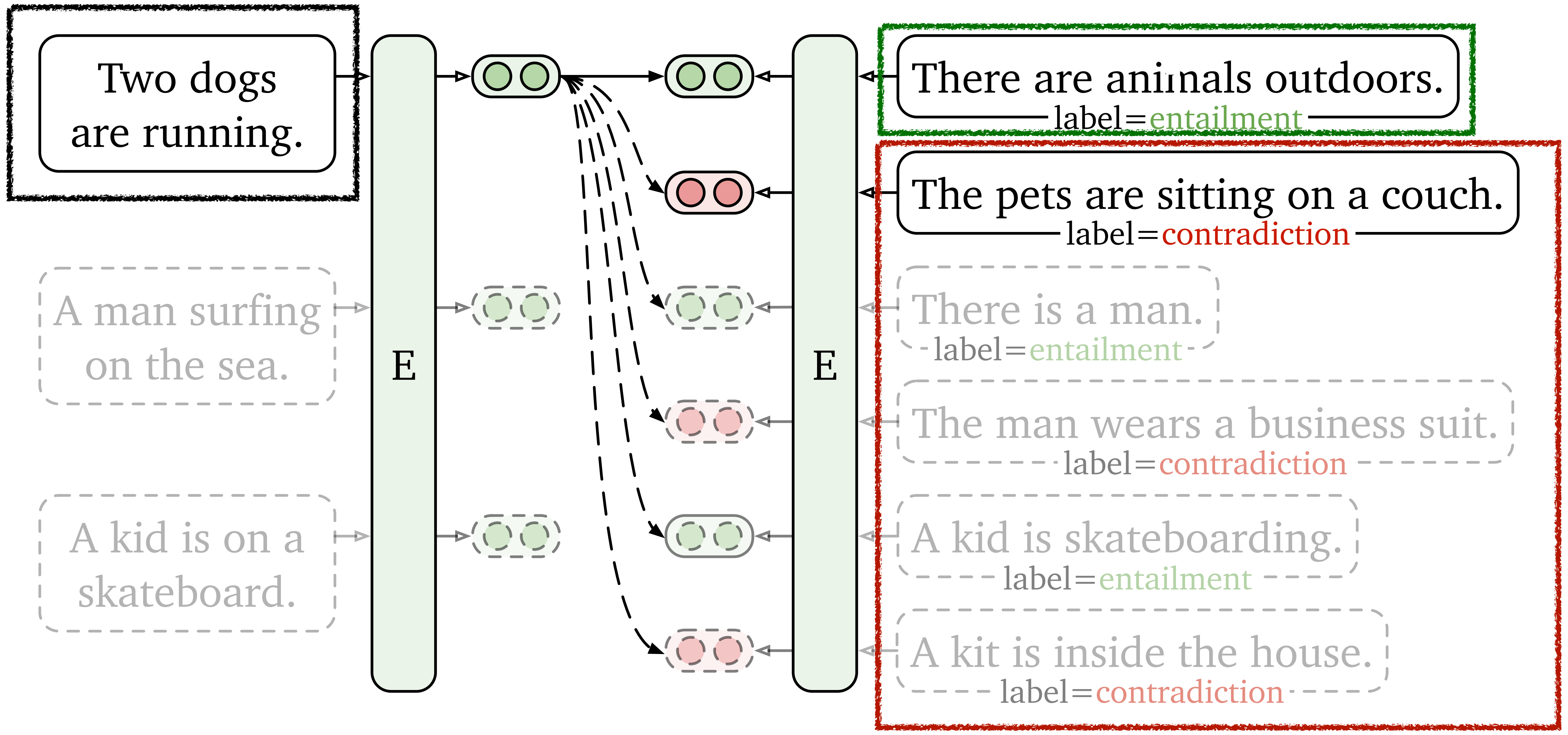
Hard negatives

SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018)

# Supervised SimCSE

**Positive pairs: entailment** (premise, hypothesis) pairs

**Negative pairs: contradiction** (premise, hypothesis) pairs + in-batch negatives




Premise

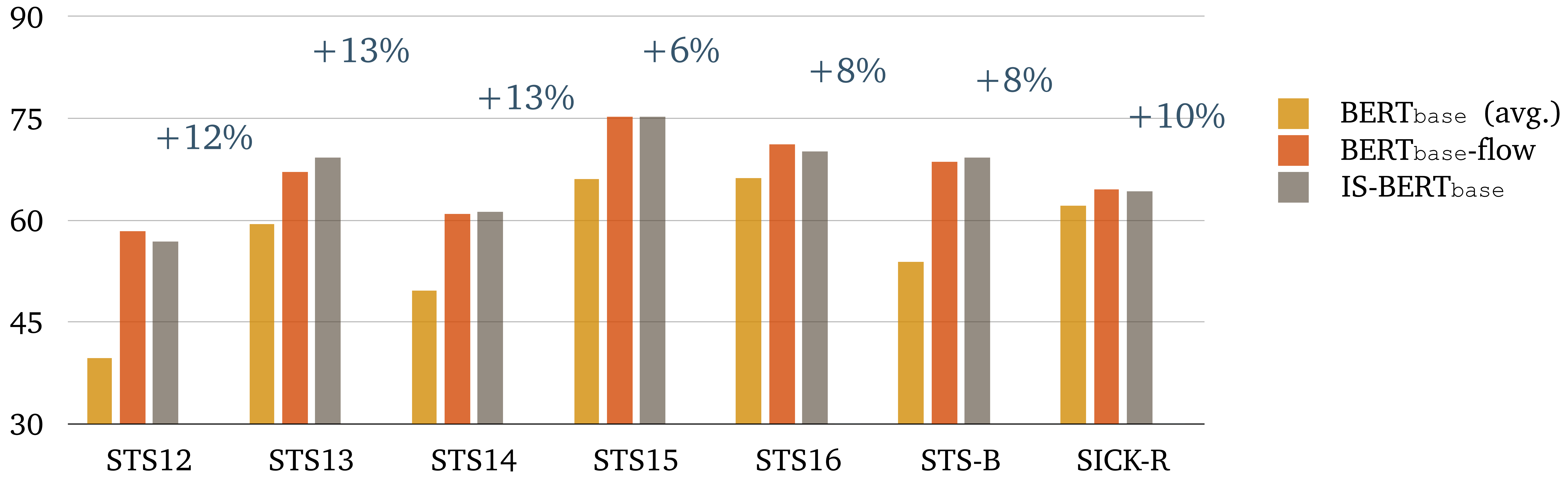
Entailment hypothesis

$$-\log \frac{e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{N}\left(e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_j^+)/\tau} + e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_j^-)/\tau}\right)}$$

Contradiction hypothesis + in-batch negatives

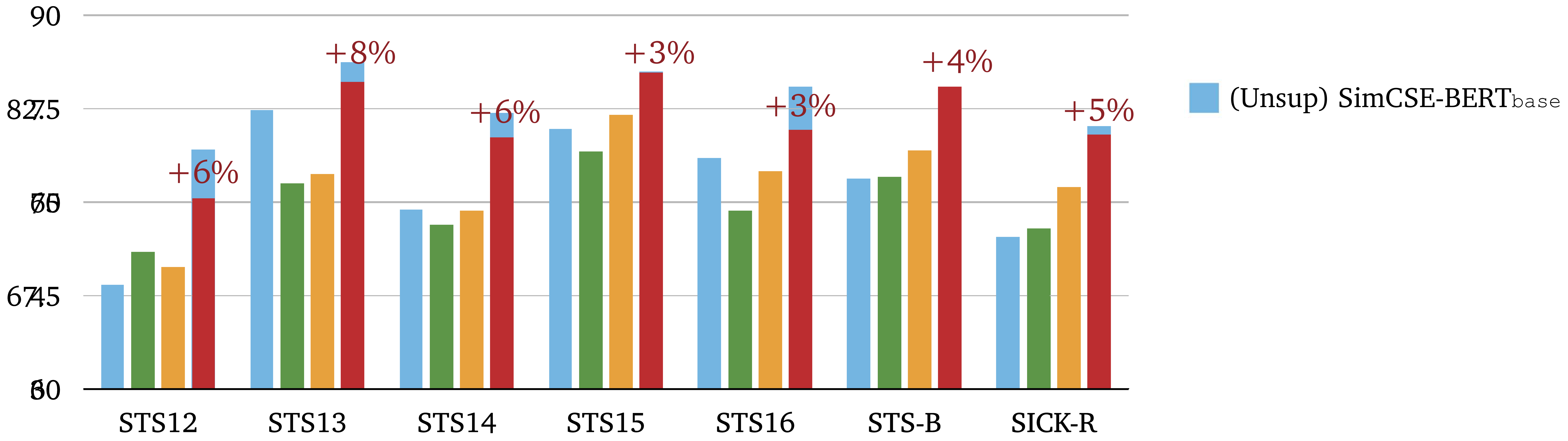
# Evaluation on STS Tasks

**Semantic textual similarity (STS)** tasks: Spearman's correlation



# Evaluation on STS Tasks

**Semantic textual similarity (STS)** tasks: Spearman's correlation




- Unsupervised SimCSE matches supervised SentenceBERT
- 6.7% higher than SentenceBERT using the same NLI datasets

(See more SentEval results in the paper)

# Why does SimCSE work?

Using **dropout masks** to create positive pairs is much better than:

- Predicting next sentences
- Discrete data augmentation (synonym/MLM replacement, word deletion, cropping)

The movie is great. vs The movie is fantastic.

~~Two dogs~~ are running. vs ~~Two~~ dogs are ~~running.~~

Two dogs are ~~running~~. vs Two dogs ~~are~~ running.

# Why does SimCSE work?

Using **dropout masks** to create positive pairs is much better than:

- Predicting next sentences
- Discrete data augmentation (synonym/MLM replacement, word deletion, cropping)

| **Training objective** | $f_\theta$ |
|---|---|
| Next sentence | 67.1 |
| Next 3 sentences | 67.4 |
| Delete one word | 75.9 |
| Unsupervised SimCSE | **82.5** |

(See more results in the paper)

more data augmentation

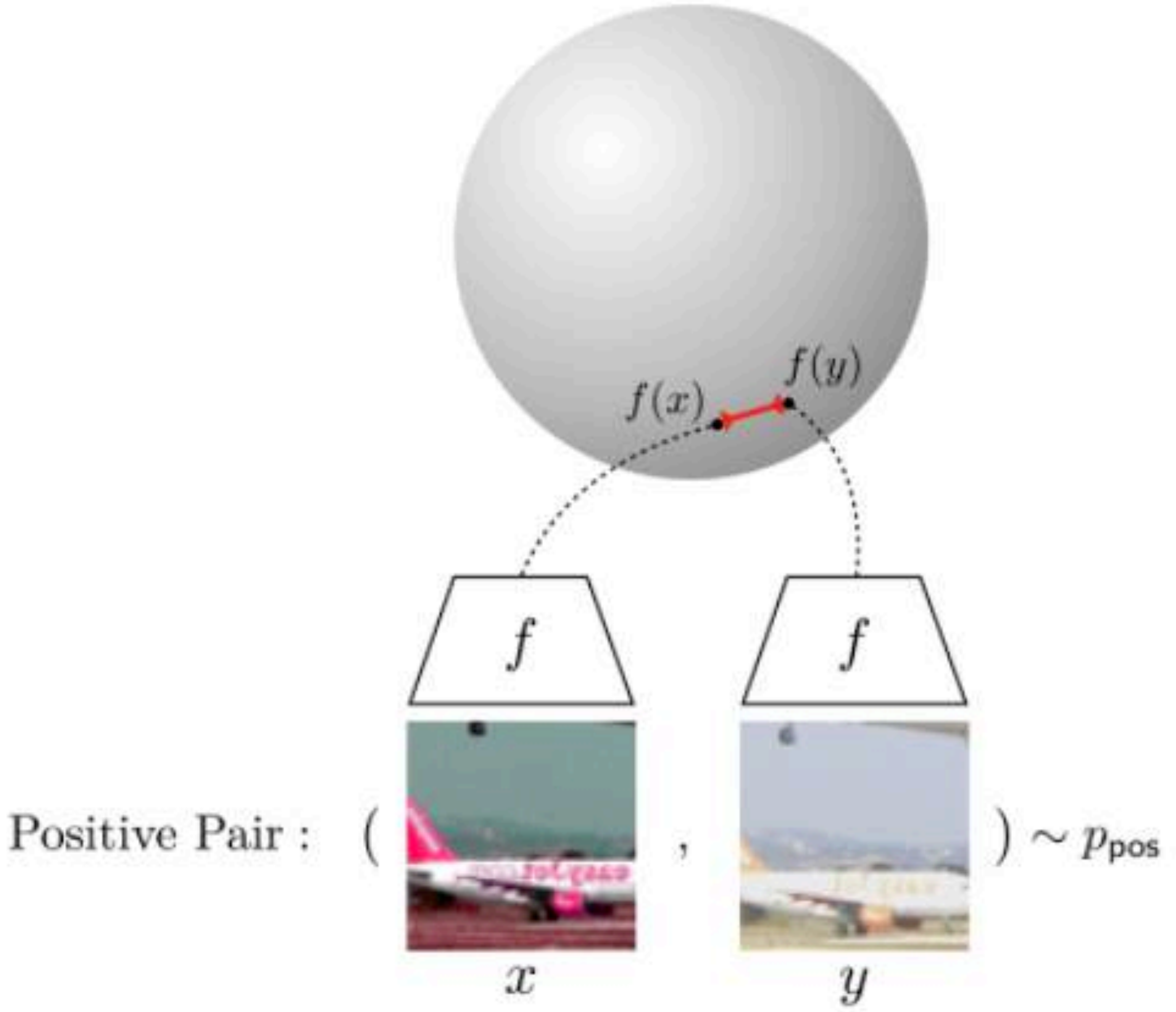
| **Data augmentation** | | | STS-B |
|---|---|---|---|
| None (unsup. SimCSE) | | | **82.5** |
| Crop | *10%* | *20%* | *30%* |
| | 77.8 | 71.4 | 63.6 |
| Word deletion | *10%* | *20%* | *30%* |
| | 75.9 | 72.2 | 68.2 |
| Delete one word | | | 75.9 |
| Synonym replacement | | | 77.4 |
| MLM 15% | | | 62.2 |

Default setting: 1 million sentences randomly sampled from English Wikipedia, N=64, evaluated on STS-B development set (Spearman correlation)

# Alignment vs uniformity

$$\ell_{\text{align}} \triangleq \mathop{\mathbb{E}}_{(x,x^+)\sim p_{\text{pos}}} \|f(x) - f(x^+)\|^2$$

$$\ell_{\text{uniform}} \triangleq \log \mathop{\mathbb{E}}_{x,y \overset{i.i.d.}{\sim} p_{\text{data}}} e^{-2\|f(x)-f(y)\|^2}$$




**Alignment:** Similar samples have similar features


**Uniformity:** Preserve maximal information

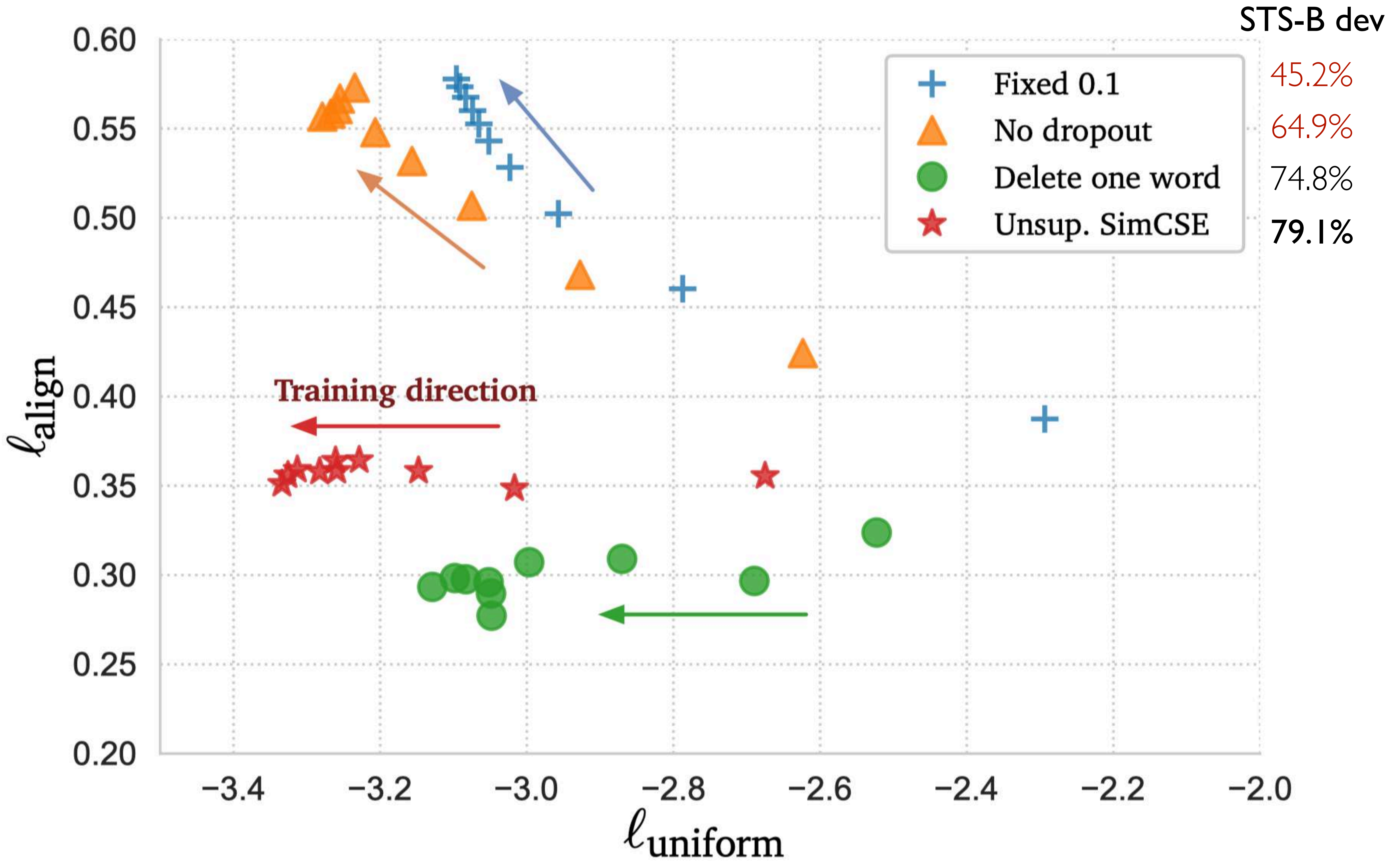
(Wang and Isola, 2020)

Alignment = how well positive pairs are aligned
Uniformity = how well the embeddings are uniformly distributed

# Alignment vs uniformity

Q: Why does different dropout masks work so well?

- Fixed 0.1
  - Standard dropout (rate=0.1)
  - Same dropout mask as positives
- No dropout
  - Dropout rate=0



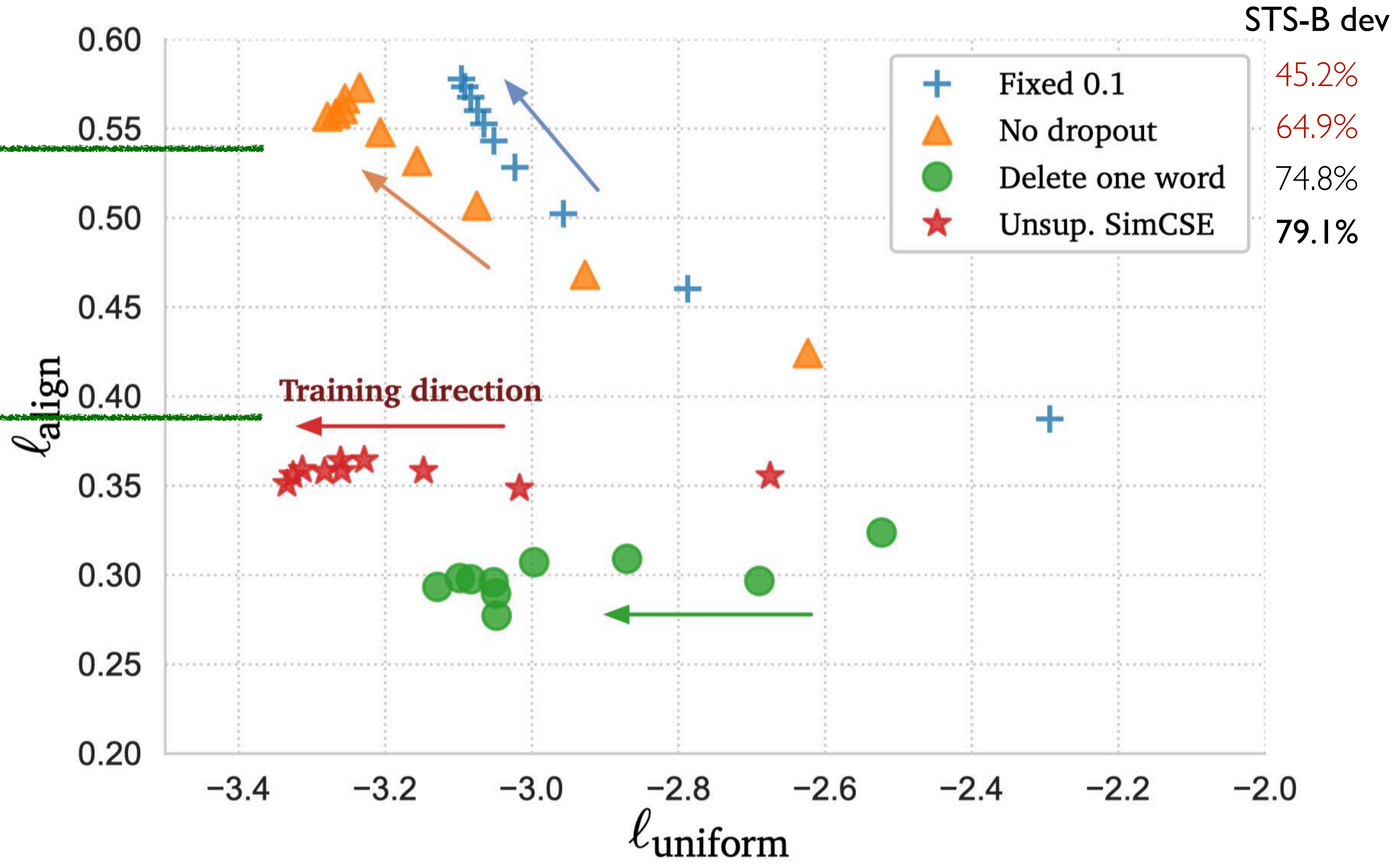

$l_{\text{uniform}}, l_{\text{align}}$: the lower, the better

# Alignment vs uniformity

Q: Why does different dropout masks work so well?

Collapse on alignment

Improve the uniformity while keeping good alignment




$l_{\text{uniform}}, l_{\text{align}}$: the lower, the better

# Why NLI datasets?

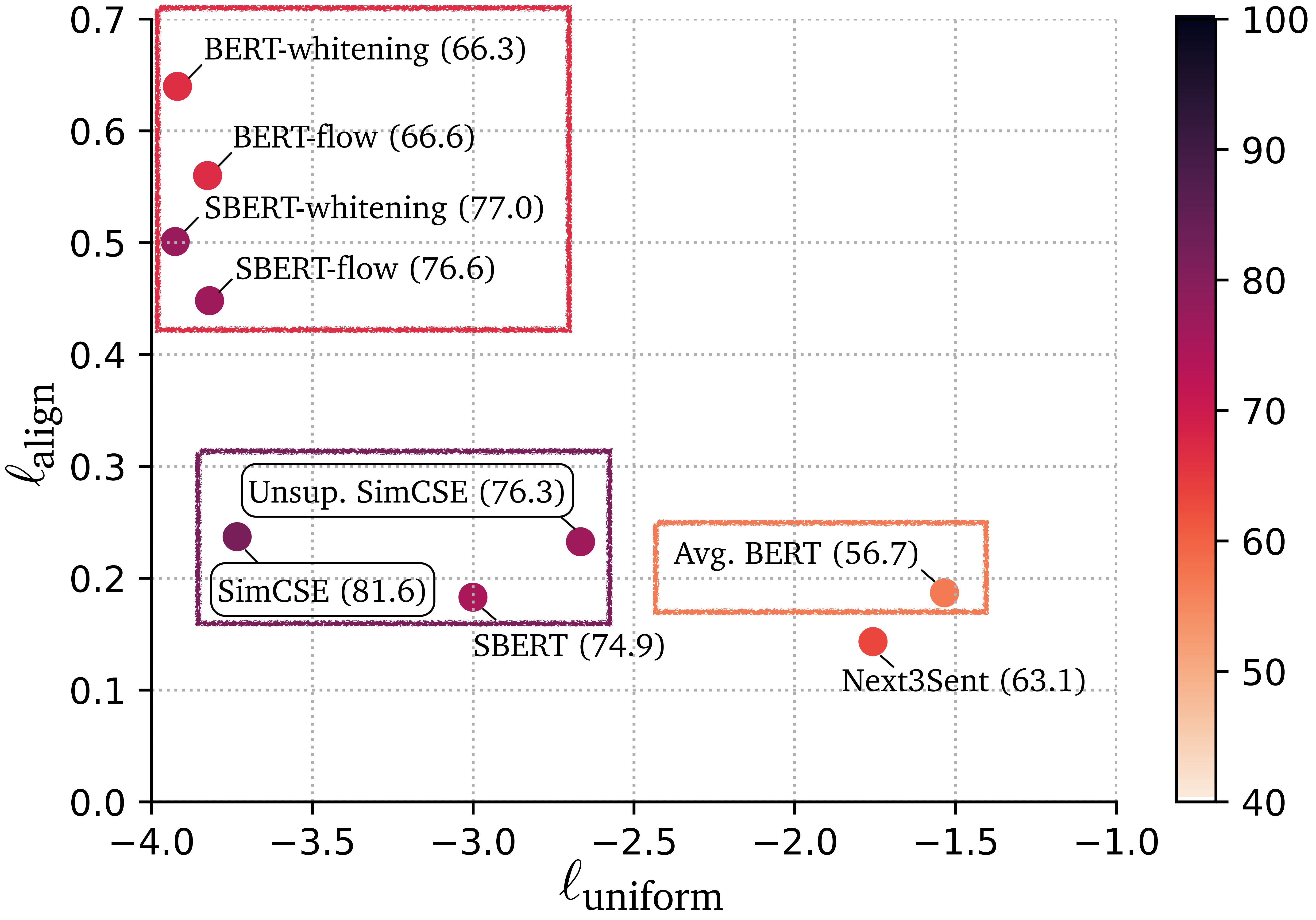
Downsampled to 134k pairs
for fair comparison

| Dataset | sample | full |
|---|---|---|
| Unsup. SimCSE (1m) | - | 79.1 |
| QQP (134k) | 81.8 | 81.8 |
| Flickr30k (318k) | 81.5 | 81.4 |
| ParaNMT (5m) | 79.7 | 78.7 |
| SNLI+MNLI | | |
| entailment (314k) | **84.1** | **84.9** |
| neutral (314k)[3] | 82.6 | 82.9 |
| contradiction (314k) | 77.5 | 77.6 |
| SNLI+MNLI | | |
| entailment + hard neg. | - | **86.2** |
| + ANLI (52k) | - | 85.0 |

No hard
negatives

**Hypothesis**: high annotation quality and small lexical overlap between pairs of sentences

# Comparison: alignment & uniformity




We also theoretically show that contrastive objective can improve the isotropy by inherently flattening the singular value distribution of the embedding space (see the paper).

$l_{\text{uniform}}, l_{\text{align}}$: the lower, the better

# Take-aways

- Contrastive learning can ease the anistropy problem (a well-known issue in pre-trained BERT representations).
- Supervised signals can better align semantically close pairs.
- Data augmentation in the continuous space is promising in NLP!
- We don't need a large batch size in learning sentence representations.

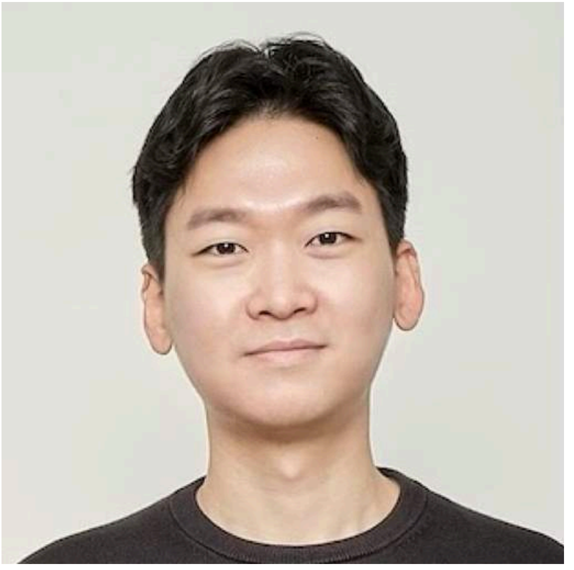
**Unsupervised SimCSE**

| **Batch size** | 32 | 64 | 128 | 256 | 512 | 1024 |
|---|---|---|---|---|---|---|
| **STS-B** | 84.6 | 85.6 | 86.0 | **86.2** | **86.2** | 86.0 |

$N = 64$ is already very good

# DensePhrases: Learning Dense Representations for Phrase Retrieval

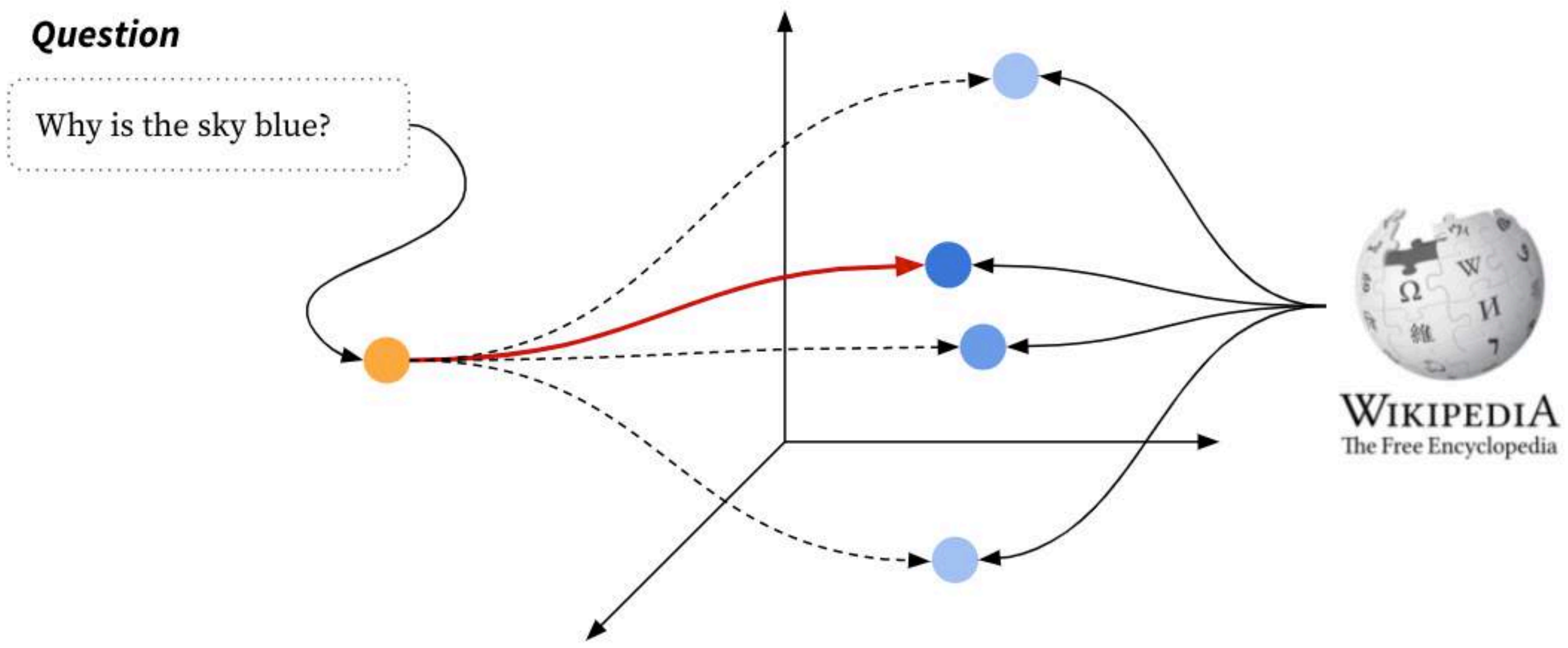
(Work done by Jinhyuk Lee, Mujeen Sung, Alexander Wettig)



https://github.com/princeton-nlp/DensePhrases

# Dense retrieval

- Encode a large collection of documents (e.g., Wikipedia) as a set of low-dimensional (e.g., 768) vectors
- Support (approximate) nearest neighbor search in this vector space
- Applications: search, open-domain QA, information extraction, fact checking, dialogue..
- Depending on retrieval unit:
  - Passage: Dense passage retriever (DPR) (Karpukhin et al., 2020)
  - Phrase: DensePhrases (Lee et al., 2021a)



# Dense passsage retrieval

Who sings Don't Stand So Close to Me? → **Question encoder**

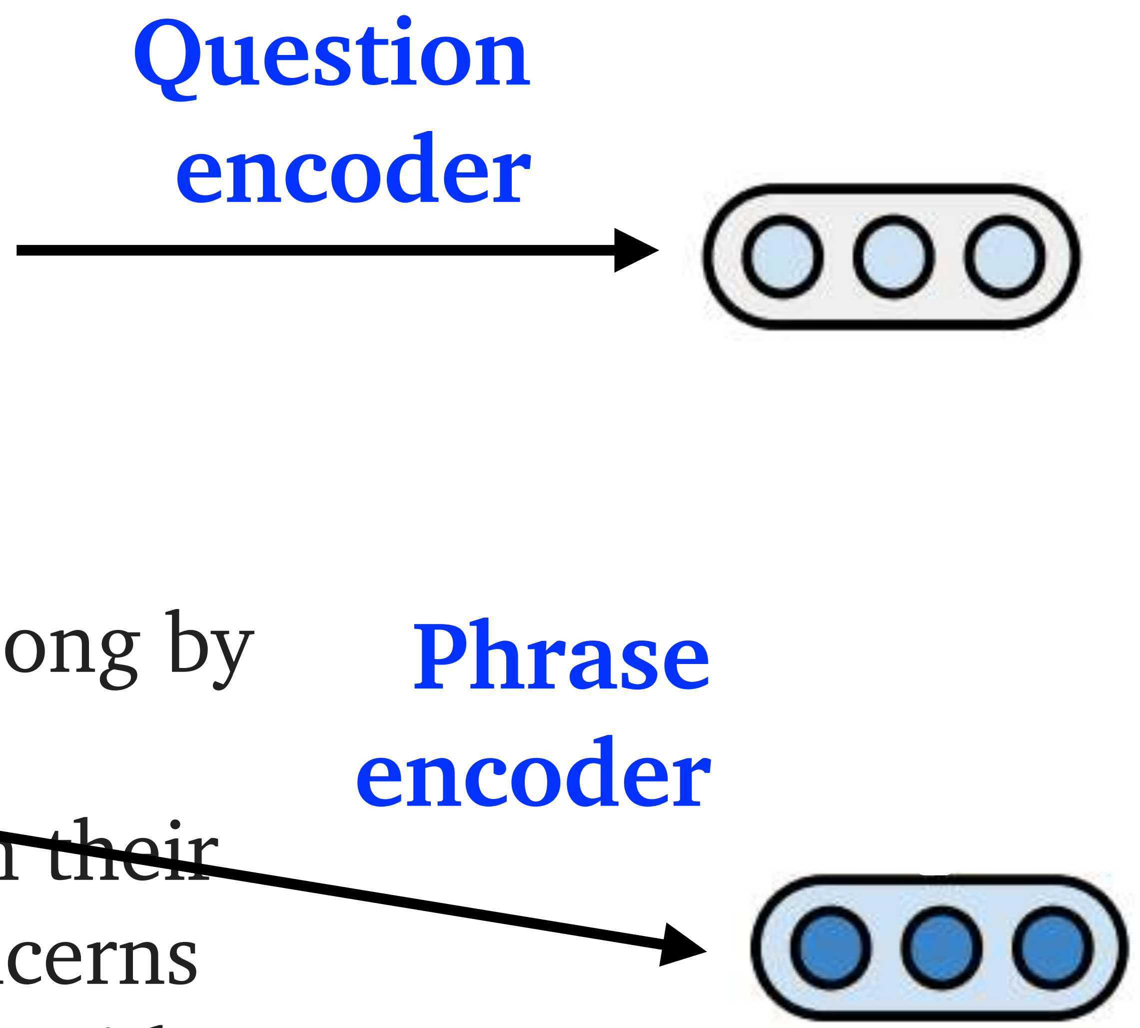
" Don't Stand So Close to Me " is a hit song by the rock band The Police , released in September 1980 as the lead single from their third album *Zenyatta Mondatta* . It concerns a teacher who has a sexual relationship with a student , which in turn is discovered .

→ **Passage encoder**

≈ 20-million passages
(offline indexing)

# Dense phrase retrieval

Who sings Don't Stand So Close to Me?




" Don't Stand So Close to Me " is a hit song by the rock band The Police , released in September 1980 as the lead single from their third album *Zenyatta Mondatta* . It concerns a teacher who has a sexual relationship with a student , which in turn is discovered .

≈ 60-billion passages (offline indexing)

- Phrase = any contiguous segment of text up to L (e.g., 20) words, NOT necessarily linguistic phrases
- All the phrases are **contextual**, e.g., there are many "The Police" phrases with different contexts

# Phrase vs passage retrieval

Retriever-reader models

(Chen et al., 2017; Lee et al., 2019; Karpukhin et al., 2020; Izacard and Grave, 2021)

Don't Stand So Close to Me

From Wikipedia, the free encyclopedia

*For the Grey's Anatomy episode, see Don't Stand So Close to Me (Grey's Anatomy).*

**"Don't Stand So Close to Me"** is a hit song by the rock band The Police, released in September 1980 as the lead single from their third album *Zenyatta Mondatta*. It concerns a schoolgirl's crush on her teacher which leads to an affair, which in turn is discovered.

T5

"The Police"

Phrase-retrieval models

"The Police"

# Phrase vs passage retrieval

| Category | Model | Sparse? | Storage (GB) | #Q/sec (GPU, CPU) | NQ (Acc) |
|---|---|---|---|---|---|
| Retriever-Reader | DrQA (Chen et al., 2017) | ✓ | 26 | 1.8, 0.6 | - |
| | BERTSerini (Yang et al., 2019) | ✓ | 21 | 2.0, 0.4 | - |
| | ORQA (Lee et al., 2019) | ✗ | 18 | 8.6, 1.2 | 33.3 |
| | $\text{REALM}_{\text{News}}$ (Guu et al., 2020) | ✗ | 18 | 8.4, 1.2 | 40.4 |
| | DPR-multi (Karpukhin et al., 2020) | ✗ | 76 | 0.9, 0.04 | 41.5 |
| Phrase Retrieval | DenSPI (Seo et al., 2019) | ✓ | 1,200 | 2.9, 2.4 | 8.1 |
| | DenSPI + Sparc (Lee et al., 2020) | ✓ | 1,547 | 2.1, 1.7 | 14.5 |
| | **DensePhrases (Ours)** | ✗ | 320 | 20.6, 13.6 | 40.9 |

New
80GB

(Lee et al., 2021a)

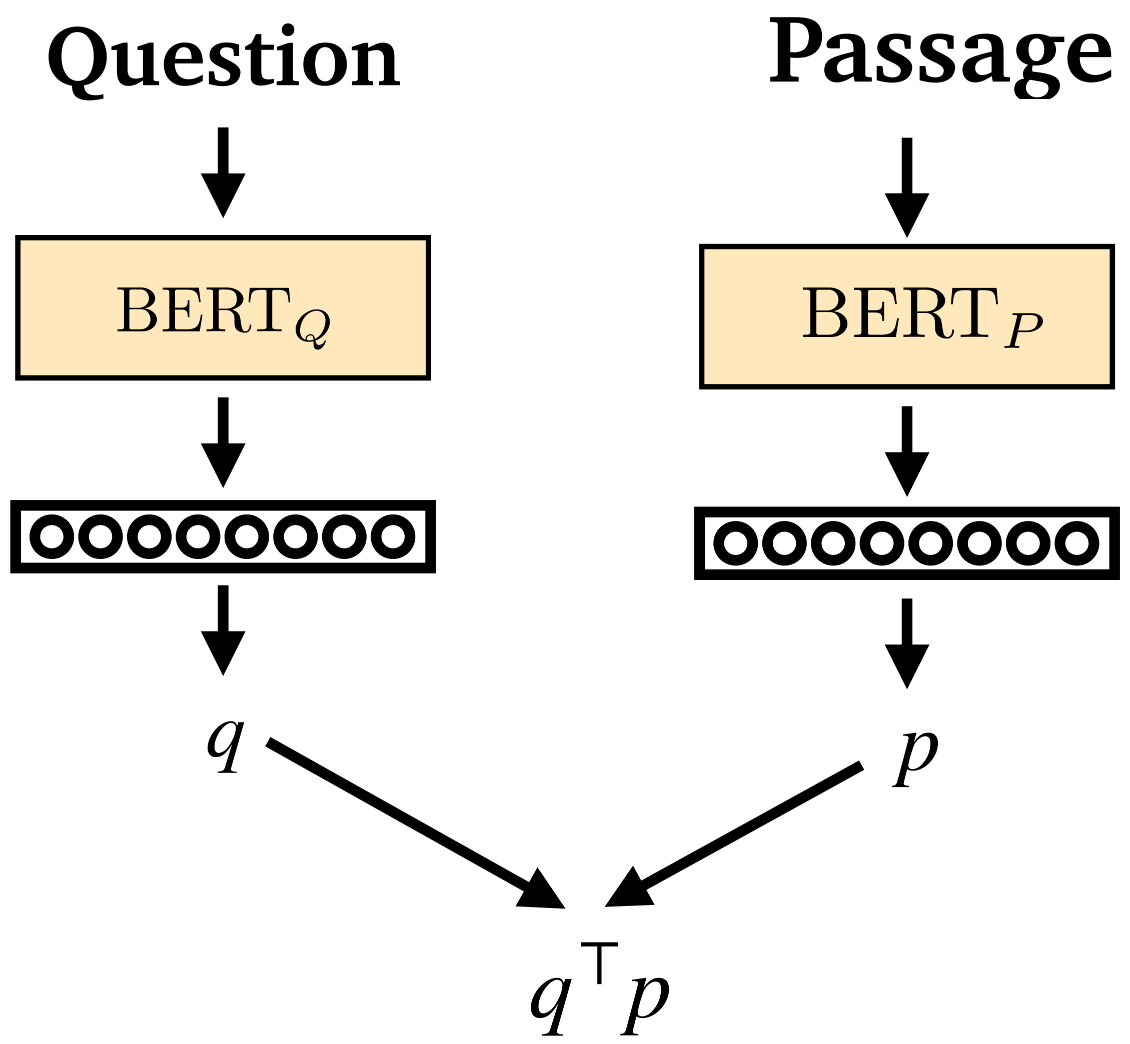
Similar accuracy

Similar storage

Much faster speed

# How to learn representations?

- **Contrastive learning** with supervised pairs!



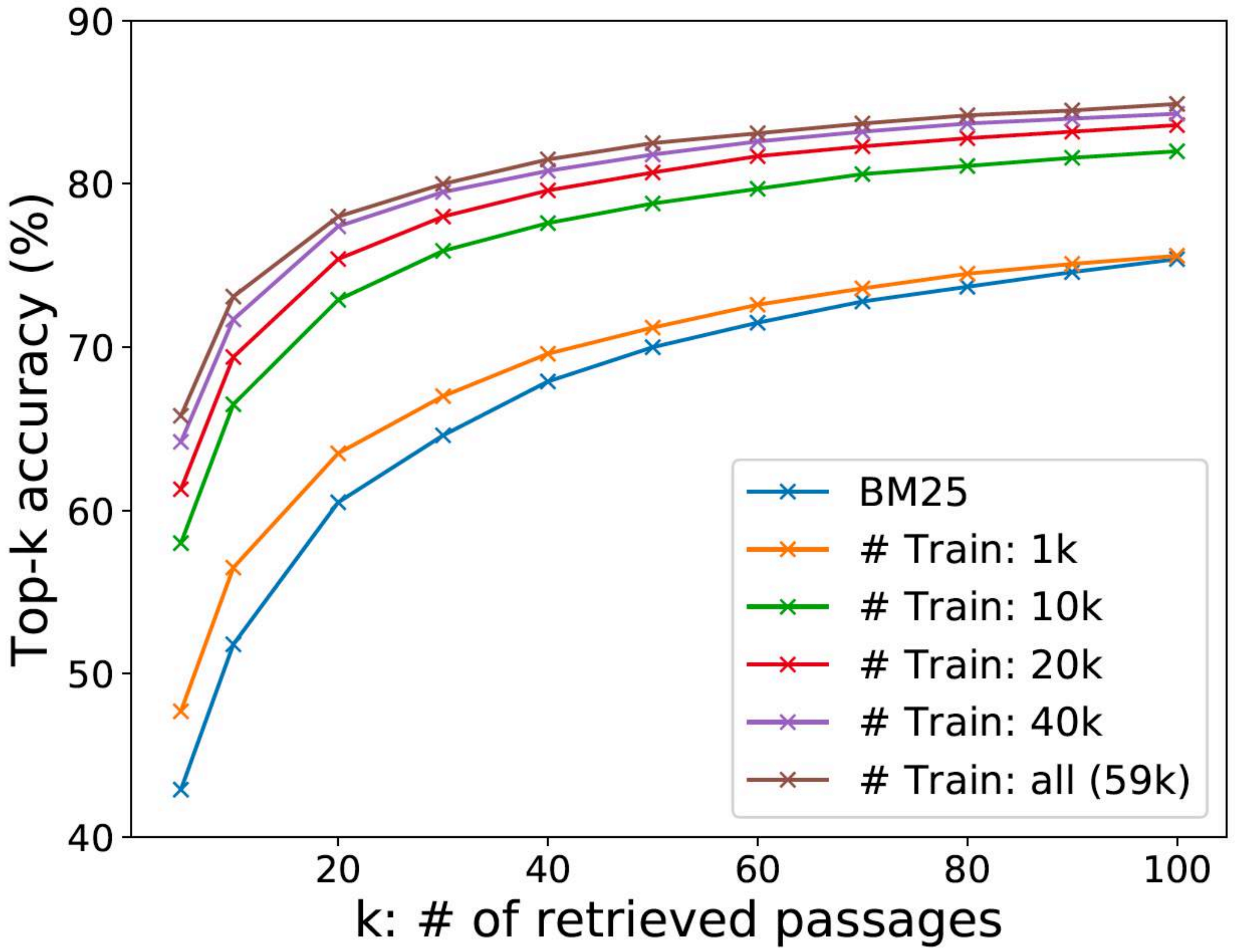

$$-\log \frac{\exp(q_i^\intercal p_i^+)}{\exp(q_i^\intercal p_i^+) + \sum_{j=1}^{N-1} \exp(q_i^\intercal p_{i,j}^-)}$$

Note: two encoders instead of one encoder!

- **Positive pairs:** (question, passage) pairs from supervised datasets
- **Negative pairs:**
  - In-batch negatives: other passages in the same mini-batch
  - Hard negatives: passages of **high BM25 scores** that do *not* contain the answer string

# DPR: positives vs negatives

**1k Q/A pairs** beat BM25!




(Karpukhin et al, 2020)

in-batch negatives

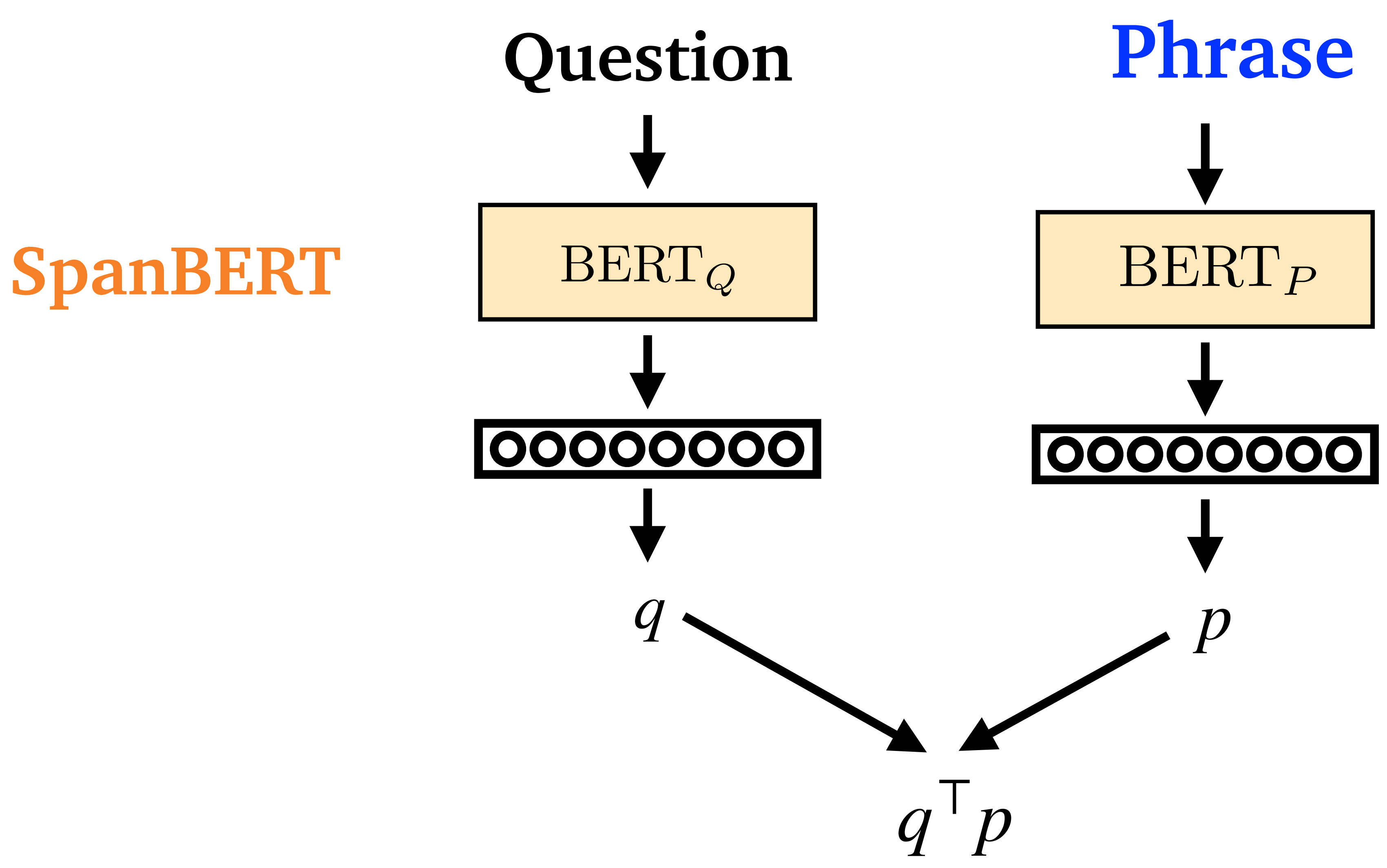
| Type | #N | IB | Top-5 | Top-20 |
|---|---|---|---|---|
| Random | 7 | ✗ | 47.0 | 64.3 |
| BM25 | 7 | ✗ | 50.0 | 63.3 |
| Gold | 7 | ✗ | 42.6 | 63.1 |
| Gold | 7 | ✓ | 51.1 | 69.1 |
| Gold | 31 | ✓ | 52.1 | 70.8 |
| Gold | 127 | ✓ | 55.8 | 73.0 |
| G.+BM25$^{(1)}$ | 31+32 | ✓ | 65.0 | 77.3 |
| G.+BM25$^{(2)}$ | 31+64 | ✓ | 64.5 | 76.4 |
| G.+BM25$^{(1)}$ | 127+128 | ✓ | **65.8** | **78.0** |

- BM25 hard negatives are important
- Batch sizes affect the performance

*: Evaluated on Natural Questions

# DensePhrases: Learning phrase representations

Very similar idea but a much harder learning task




phrase representations

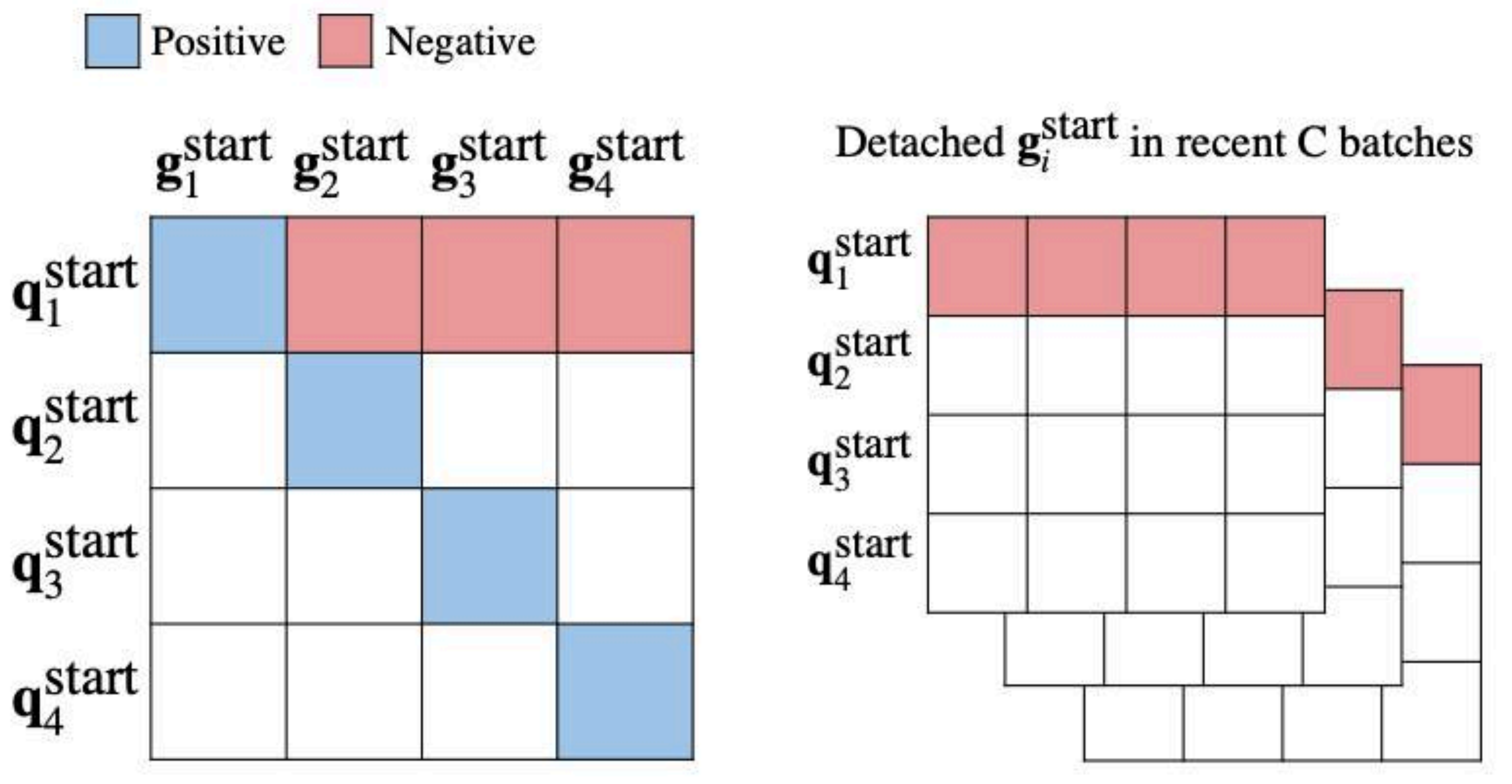
$$-\log \frac{\exp(q_i^\intercal p_i^+)}{\exp(q_i^\intercal p_i^+) + \sum_{j=1}^{N-1} \exp(q_i^\intercal p_{i,j}^-)}$$

Key ingredients:

- Batch sizes are important! We proposed **pre-batch negatives** to increase # of negatives
- The other phrases in the same passage act as **hard negatives** (= no need to use BM25 hard negatives)

# In-batch vs pre-batch negatives

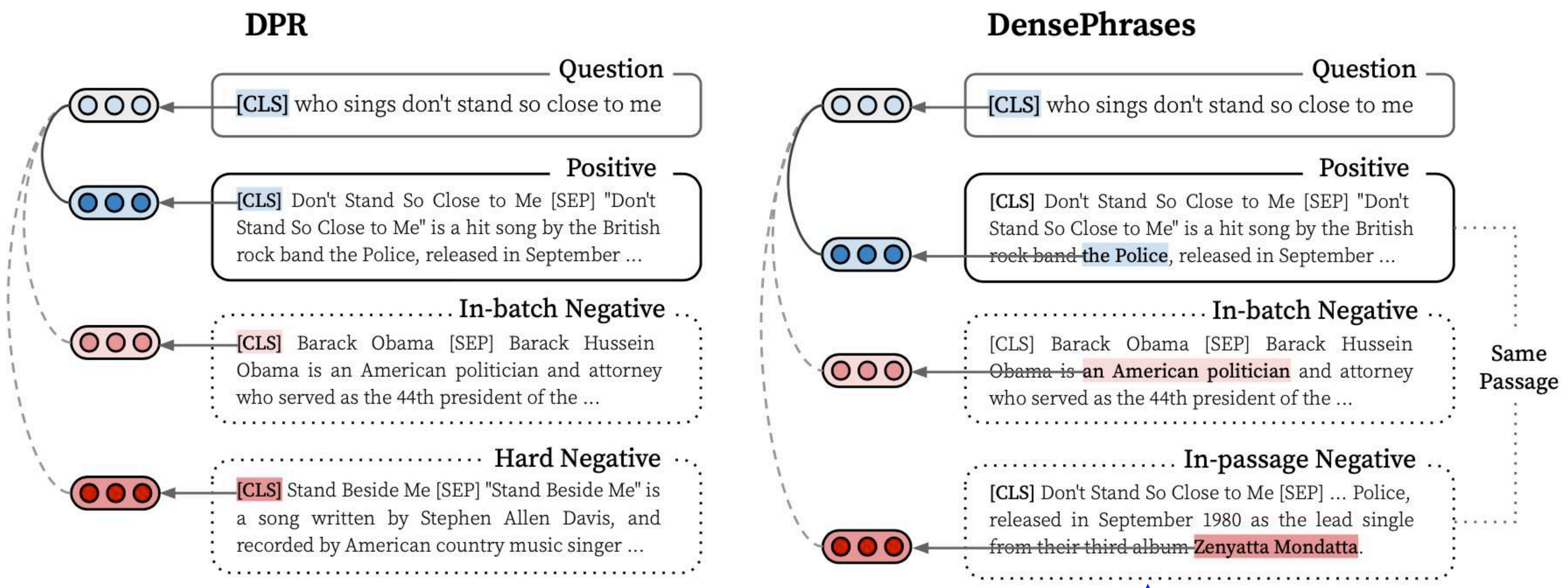
- In-batch negatives (batch size = B)
- **Pre-batch negatives:** use even more negative examples from previous batches! Build a FIFO queue and cache C previous batches, so we get B × C additional negative examples




(a) In-batch Negatives ($B-1$) (b) Pre-batch Negatives ($B \times C$)

| Type | $B$ | $C$ | $\mathcal{D} = \{p\}$ | $\mathcal{D} = \mathcal{D}_{\text{small}}$ |
|---|---|---|---|---|
| None | 48 | - | 70.4 | 35.3 |
| + In-batch | 48 | - | 70.5 | 52.4 |
| | 84 | - | 70.3 | 54.2 |
| + Pre-batch | 84 | 1 | 71.6 | 59.8 |
| | 84 | 2 | **71.9** | **60.4** |
| | 84 | 4 | 71.2 | 59.8 |

# Importance of hard negatives




BM25 hard negatives

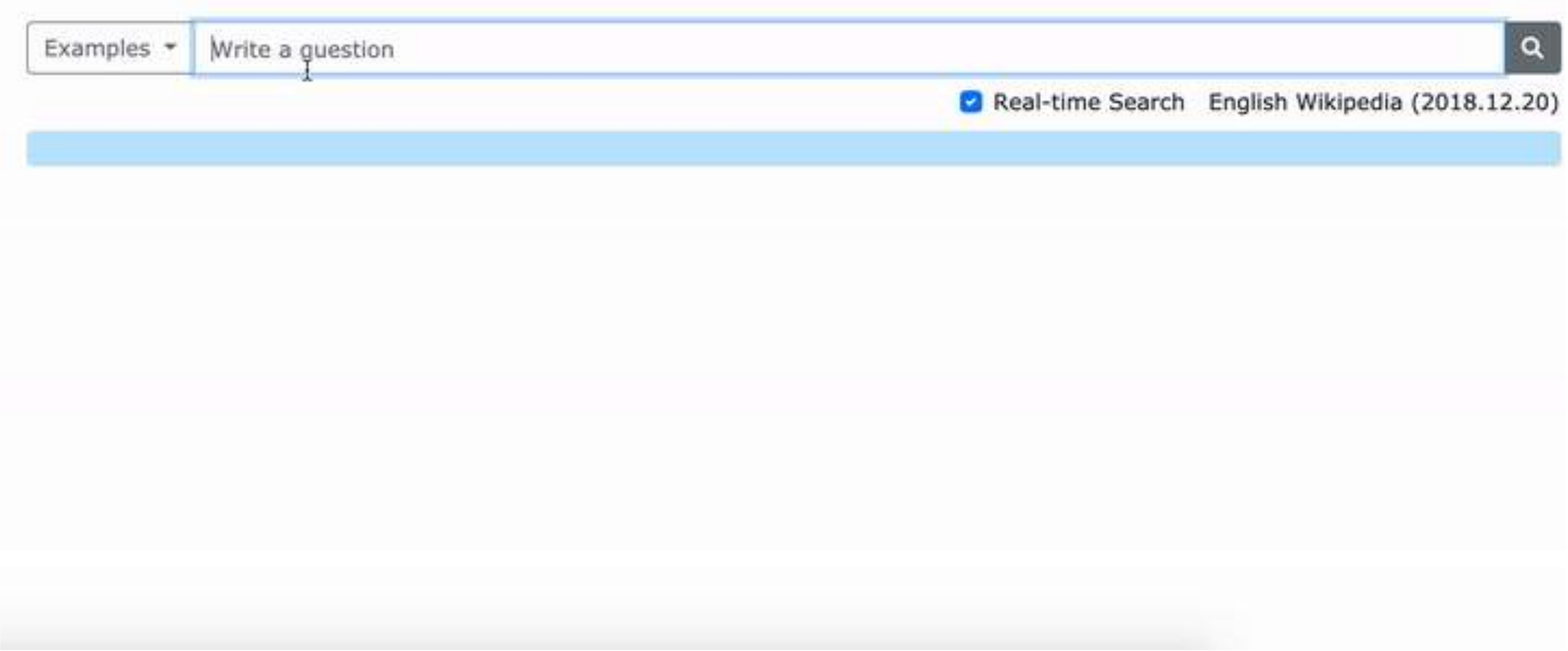
Other phrases in the same passage act as hard negatives implicitly!

(Lee et al., 2021b)

# Take-aways

- Contrastive learning can be very effective in learning dense representations for retrieval
    - **Dual-encoder** framework: both initialized from BERT
    - Positive pairs come from **supervised datasets** (even 1k pairs works!)
    - Both batch sizes and hard negatives are important

- (Not relevant to this talk) DensePhrases can support dense retrieval of different granularity in real time!



# Conclusion

$$\mathcal{L}_N = -\mathbb{E}_X \left( \log \frac{\exp(\mathrm{sim}(f(x), f(x^+)))}{\exp(\mathrm{sim}(f(x), f(x^+))) + \sum_{j=1}^{N-1} \exp(\mathrm{sim}(f(x), f(x_j)))} \right)$$

**Key ingredients:**

- Where do **positive pairs** come from (e.g., **data augmentation**)?
- The impact of batch size (= how many **negatives**)?
- Hard negatives

With pre-trained representations, contrastive learning works well in text,

- When we have the right **data augmentations**
- When we have the right **supervision** of "paired data"

# Conclusion

RQ2. Why not contrastive learning in pre-training?

Example: COCO-LM (Meng et al., 2021)


Thanks!

danqic@cs.princeton.edu