University of Chinese Academy of Sciences

INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

# Pre-train a Discriminative Text Encoder for Dense Retrieval via Contrastive Span Prediction

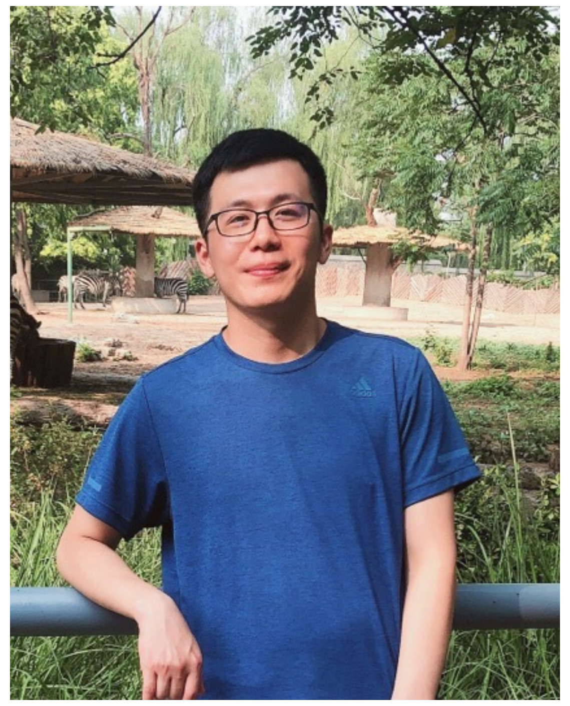
**Xinyu Ma**, Jiafeng Guo, Ruqing Zhang, Yixing Fan, and Xueqi Cheng

https://arxiv.org/pdf/2204.10641.pdf

1. CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences

2. University of Chinese Academy of Sciences

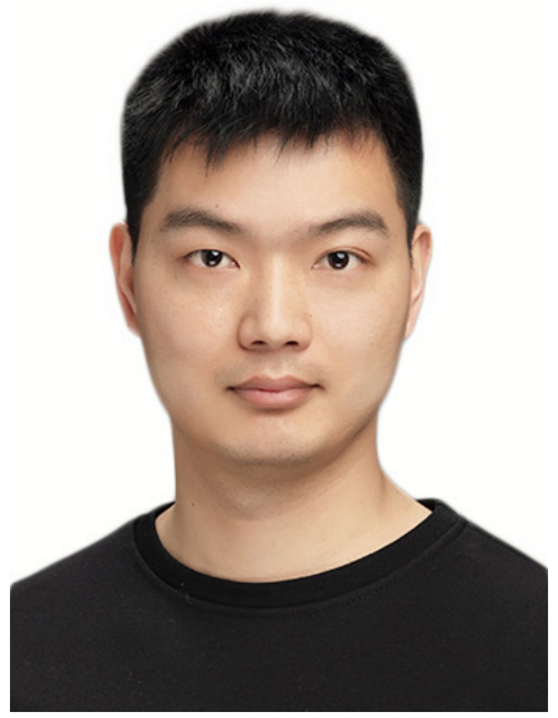
# Authors

**Xinyu Ma**
Ph.D. student
University of the Chinese Academy of Sciences

**Jiafeng Guo**
Professor
Chinese Academy of Sciences

**Ruqing Zhang**
Assistant Professor
Chinese Academy of Sciences

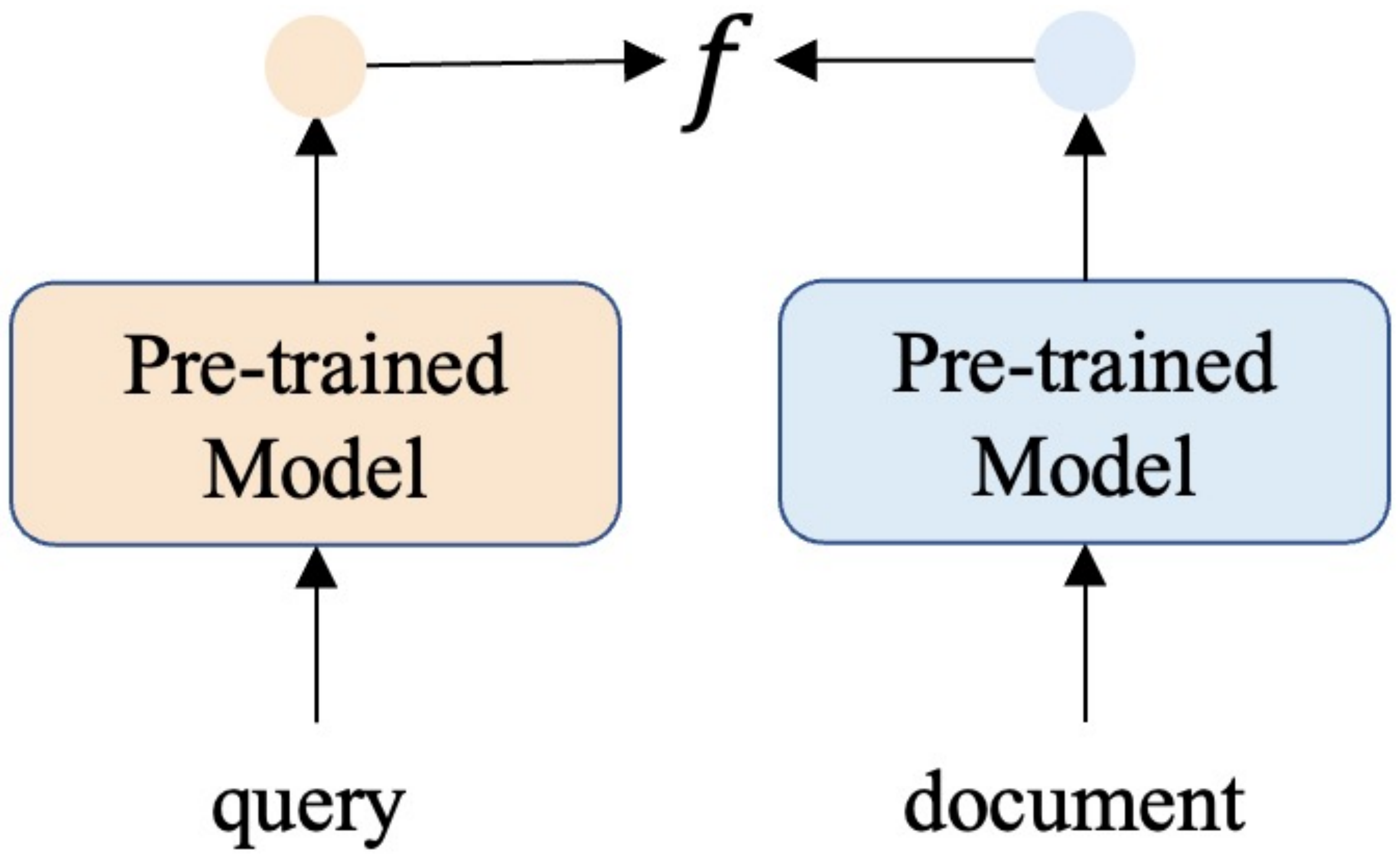


**Yixing Fan**
Associate Professor
Chinese Academy of Sciences

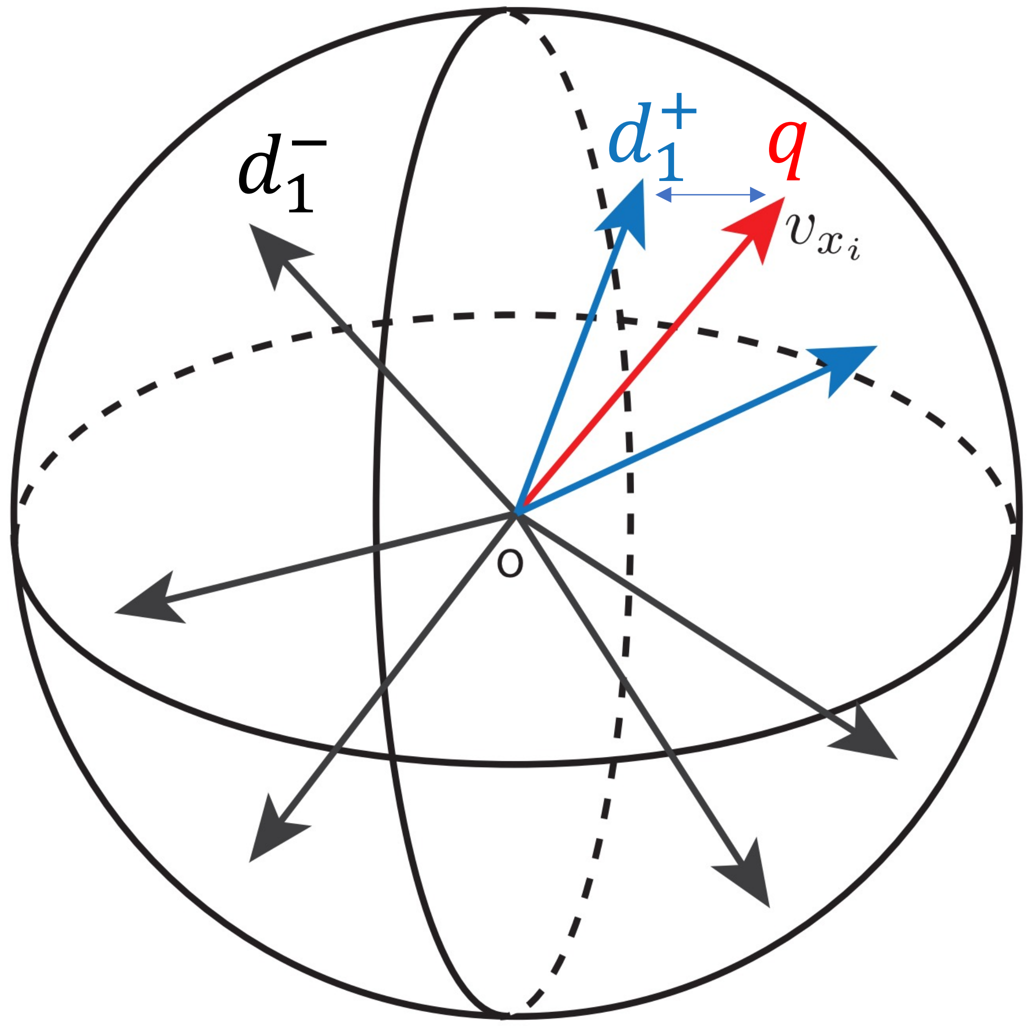
**Xueqi Cheng**
Professor
Chinese Academy of Sciences

# Dense Retrieval

- Dense retrieval has shown promising results in information retrieval (IR).




a) Model architecture




b) Search in the representation space

- The foundation of effective search is **high-quality text representation learning.**

# Recap the BERT model

- BERT learns contextualized **word** representation and inter-sequence **coherence** relationship.

| BERT Pre-training | VS | Requirements of DR |
|---|---|---|
| **Token-level** representations<br>**MLM, Language Modeling** | ✗ | **Sequence-level** representations for **short queries and long documents** |
| Sequence-level **coherence** based on the **interactions** of two **concatenated** sentence<br>**NSP, Sentence Order Tasks** | ✗ | **Relevance** relationship based on the **separated** text sequence **representations** |

- **There is still a gap between BERT and the requirements of dense retrieval**

*BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, NAACL 2019*

# The weakness of BERT

- BERT is not good at producing high-quality **text sequence representations**

| Model | STS12 | STS13 | STS14 | STS15 | STS16 | STSb | SICK-R | Avg. |
|---|---|---|---|---|---|---|---|---|
| Avg. GloVe embeddings | 55.14 | 70.66 | 59.73 | 68.25 | 63.66 | 58.02 | 53.76 | 61.32 |
| Avg. BERT embeddings | 38.78 | 57.98 | 57.98 | 63.15 | 61.06 | 46.35 | 58.40 | 54.81 |
| BERT CLS-vector | 20.16 | 30.01 | 20.09 | 36.88 | 38.08 | 16.50 | 42.63 | 29.19 |
| InferSent - Glove | 52.86 | 66.75 | 62.15 | 72.77 | 66.87 | 68.03 | 65.65 | 65.01 |
| Universal Sentence Encoder | 64.49 | 67.80 | 64.61 | 76.83 | 73.18 | 74.92 | **76.69** | 71.22 |
| SBERT-NLI-base | 70.97 | 76.53 | 73.19 | 79.09 | 74.30 | 77.03 | 72.91 | 74.89 |
| SBERT-NLI-large | 72.27 | **78.46** | **74.90** | 80.99 | 76.25 | **79.23** | 73.75 | 76.55 |
| SRoBERTa-NLI-base | 71.54 | 72.49 | 70.80 | 78.74 | 73.69 | 77.77 | 74.46 | 74.21 |
| SRoBERTa-NLI-large | **74.53** | 77.00 | 73.18 | **81.85** | **76.82** | 79.10 | 74.29 | **76.68** |

Table 1: Spearman rank correlation $\rho$ between the cosine similarity of sentence representations and the gold labels for various Textual Similarity (STS) tasks. Performance is reported by convention as $\rho \times 100$. STS12-STS16: SemEval 2012-2016, STSb: STSbenchmark, SICK-R: SICK relatedness dataset.

- The text sequence representations from original BERT is worse than GloVe.

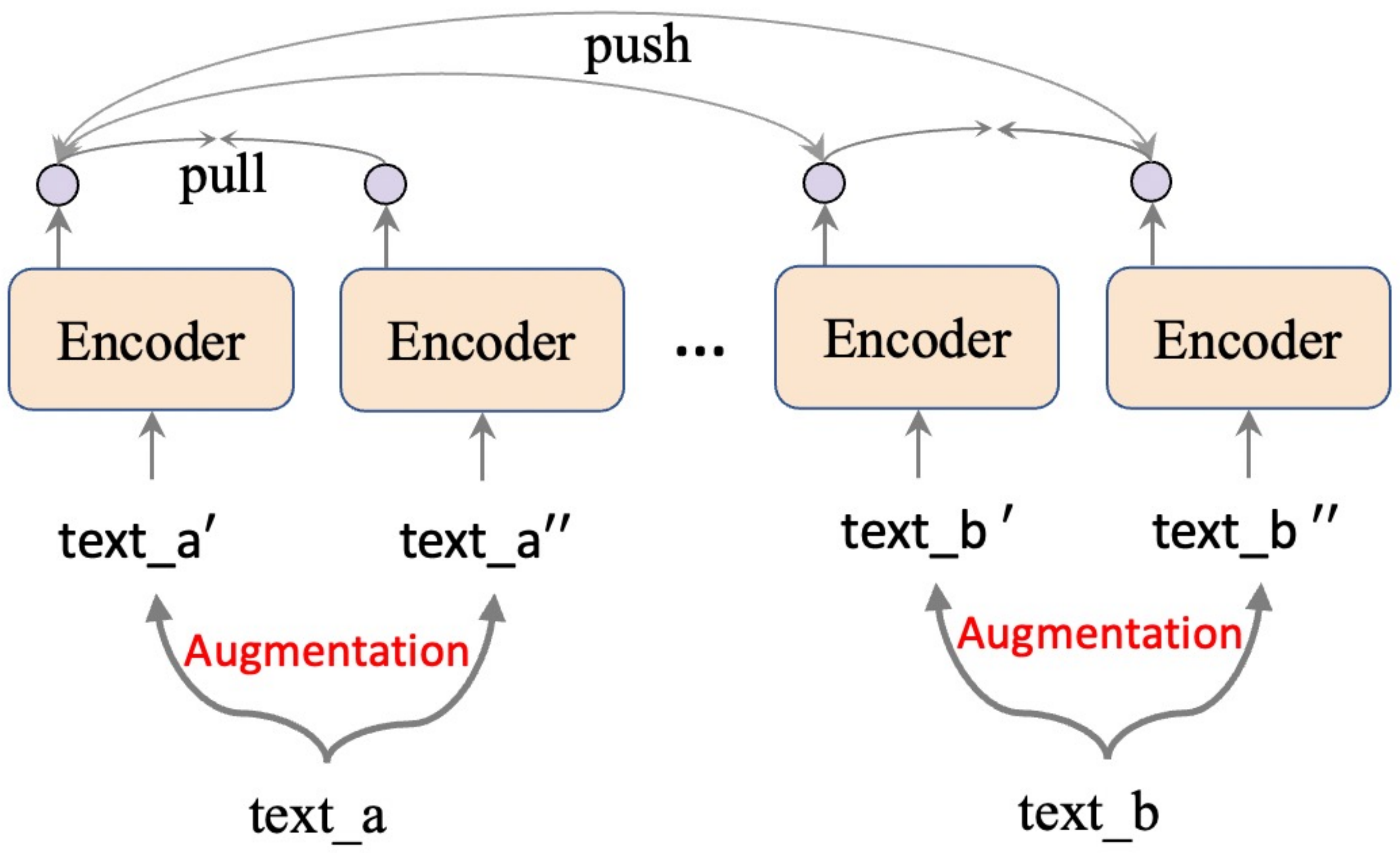
*Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, EMNLP 2019*

# Our Goal

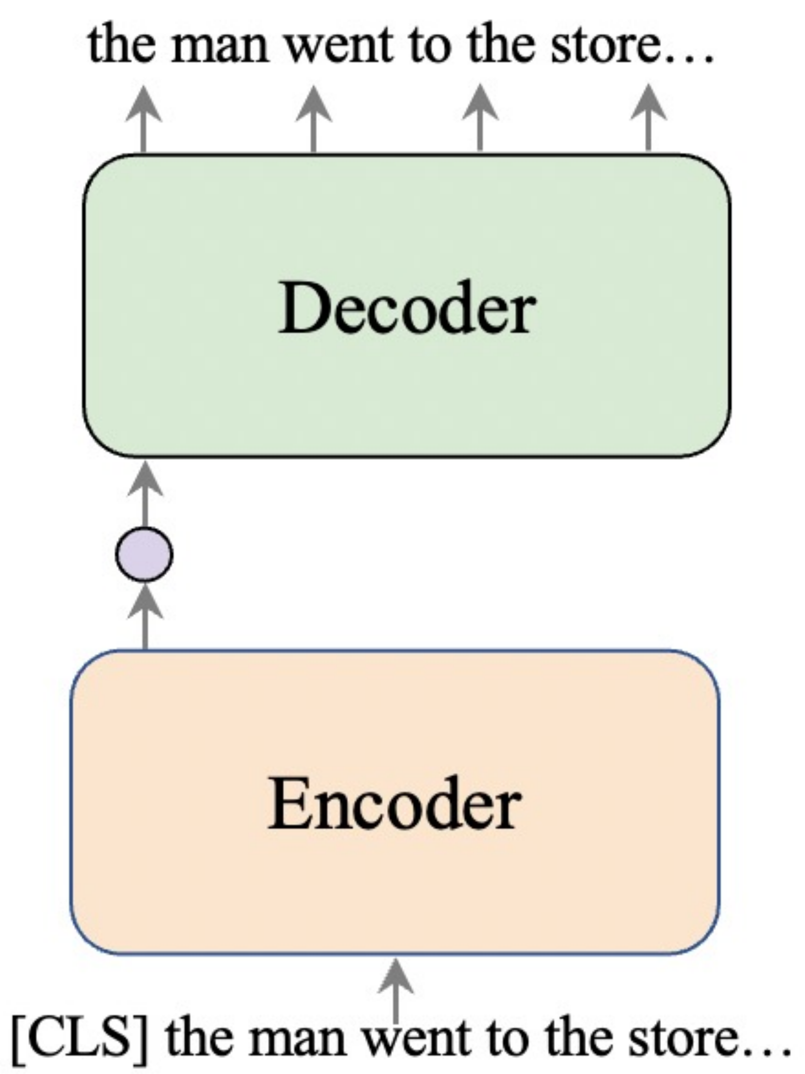
**Pre-train a discriminative text encoder tailored for dense retrieval to improve the retrieval performance and fine-tuning efficiency**

# Related Work

- Two categories: contrastive learning vs. autoencoder-based




(a) Contrastive Learning



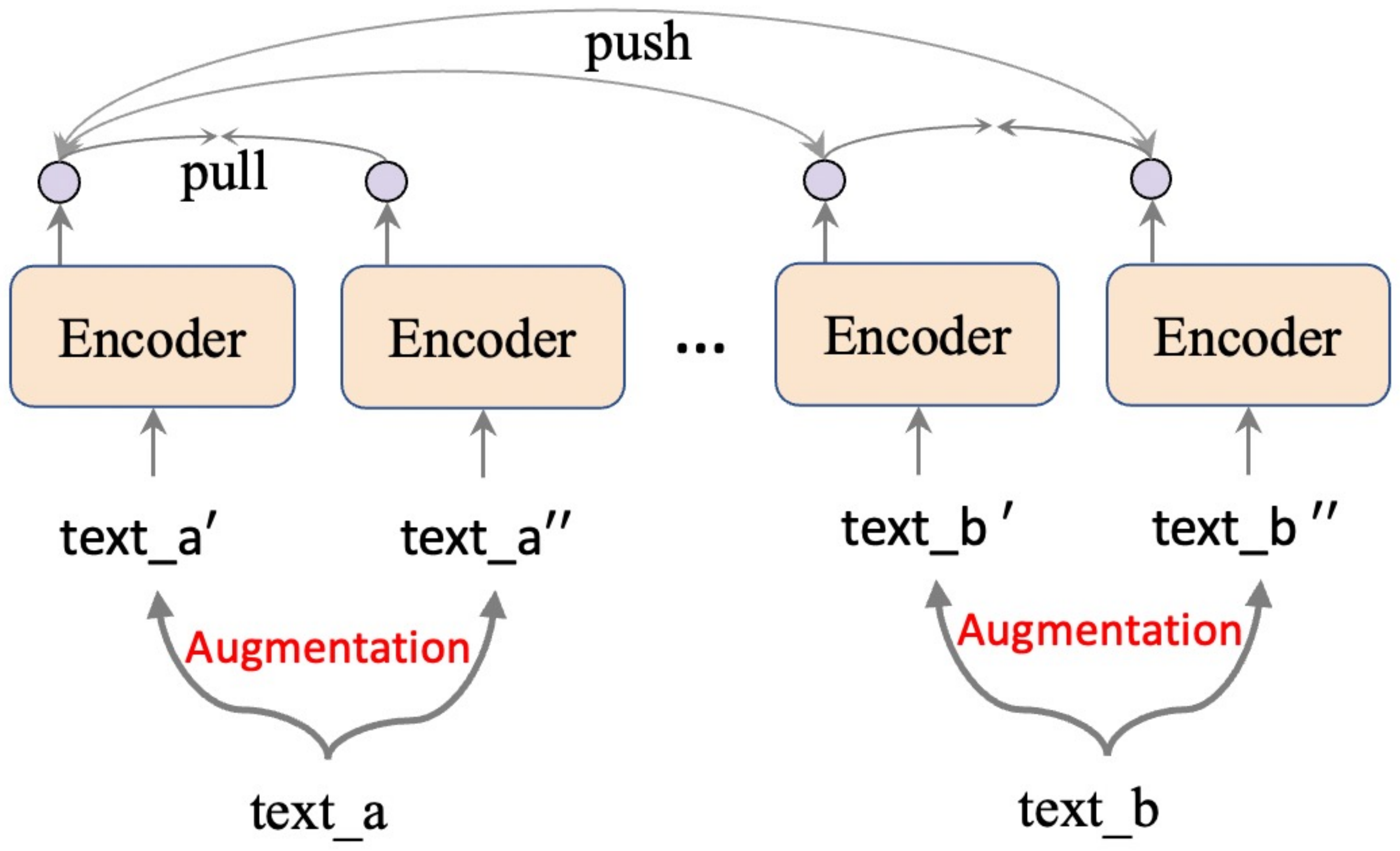

(b) Auto-encoder

- Sentence-BERT, Reimers et.al.
- SimCSE, Gao et.al.
- DeCLUTR, Giorgi et.al.
- …

- Optimus, Li et.al.
- **Seed, Lu et.al.**

# Contrastive Learning

- Pull the positive pairs in the semantic space close and push away from negatives



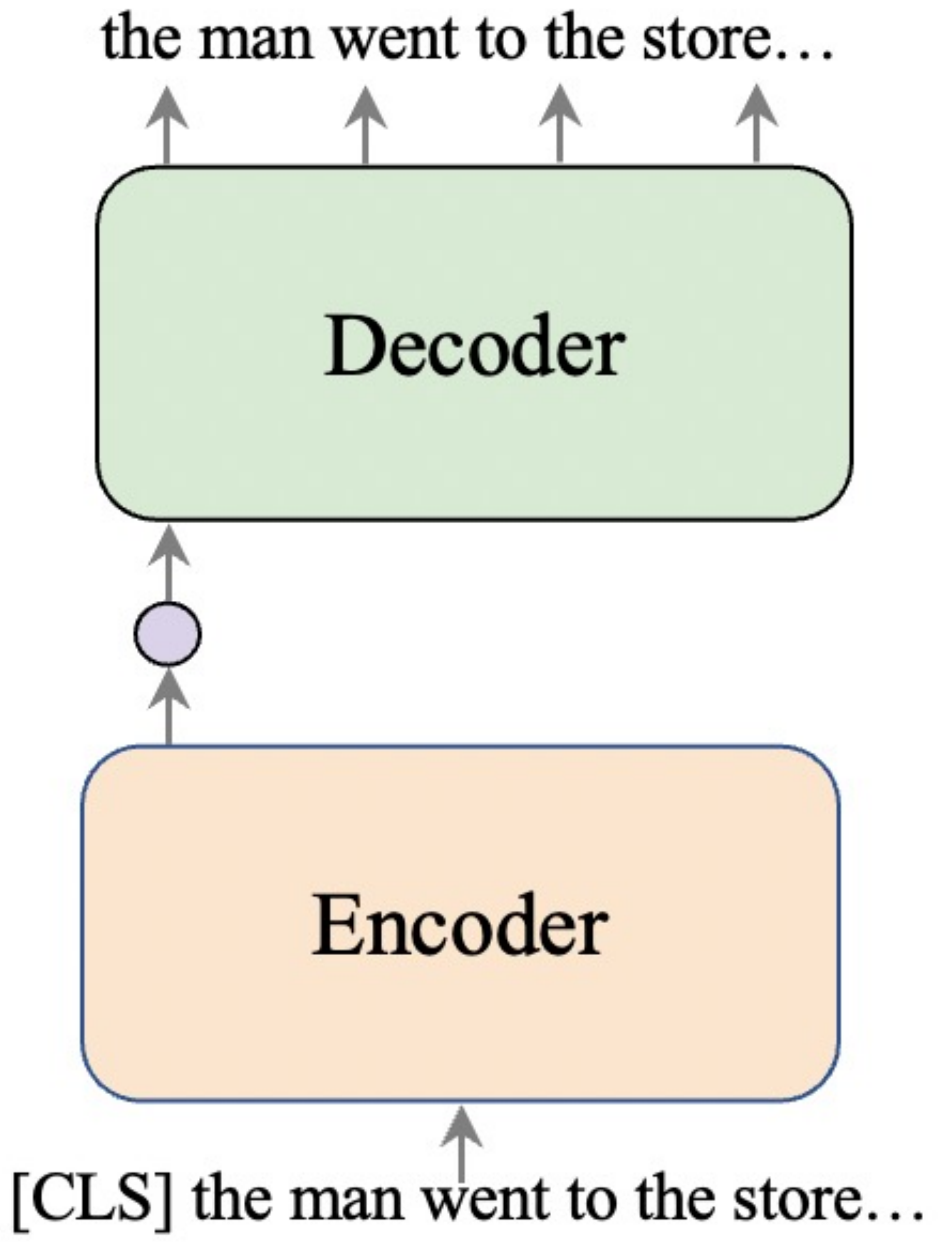
(a) Contrastive Learning

- Advantages: good **discriminative** ability
- Challenge: how to augment long text?

- Existing work:
  - Most focus on sentence-level or short passage-level, **not document-level**
  - Their augmentation methods don't work on longer text (**too easy**)

# Autoencoder-based

- Learn high-quality representations by reconstructing the input text



(b) Auto-encoder

- Advantages: create a **bottleneck**
- Challenge: not discriminative and suffer from the bypass effect
  - Treat all the tokens equally when decoding
  - Predict the next token only based on previous tokens

- Existing works: pre-train a strong encoder with a **weak** decoder to alleviate the bypass effect

*Less is More: Pre-train a Strong Text Encoder for Dense Retrieval Using a Weak Decoder, EMNLP 2021*

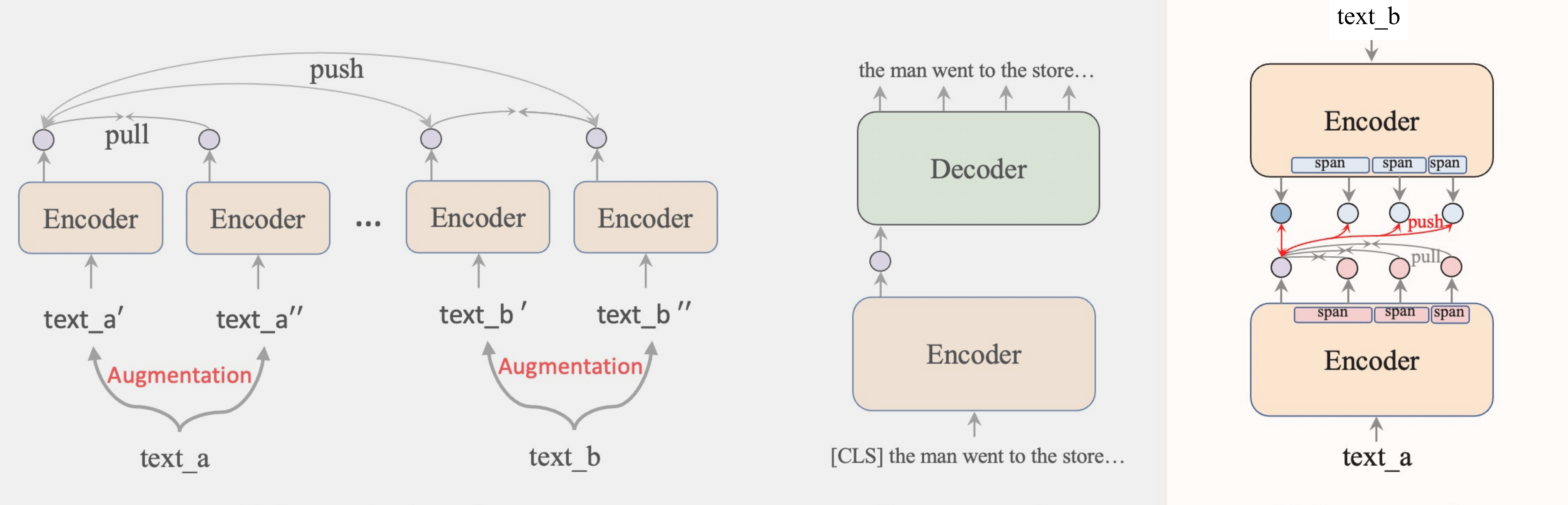
# Motivation

- Can we learn a discriminative text encoder for dense retrieval with the pros of these two methods but avoid their cons?

| | Contrastive Learning | Autoencoder-based |
|---|---|---|
| Pros | Discriminative | Bottleneck |
| Cons | Data augmentation methods designed for sentence-level are not suitable for document-level | (1) Not discriminative to decode all tokens equally; (2) The bypass effect for autoregressive decoder |

# COSTA—COntrastive Span predicTion tAsk

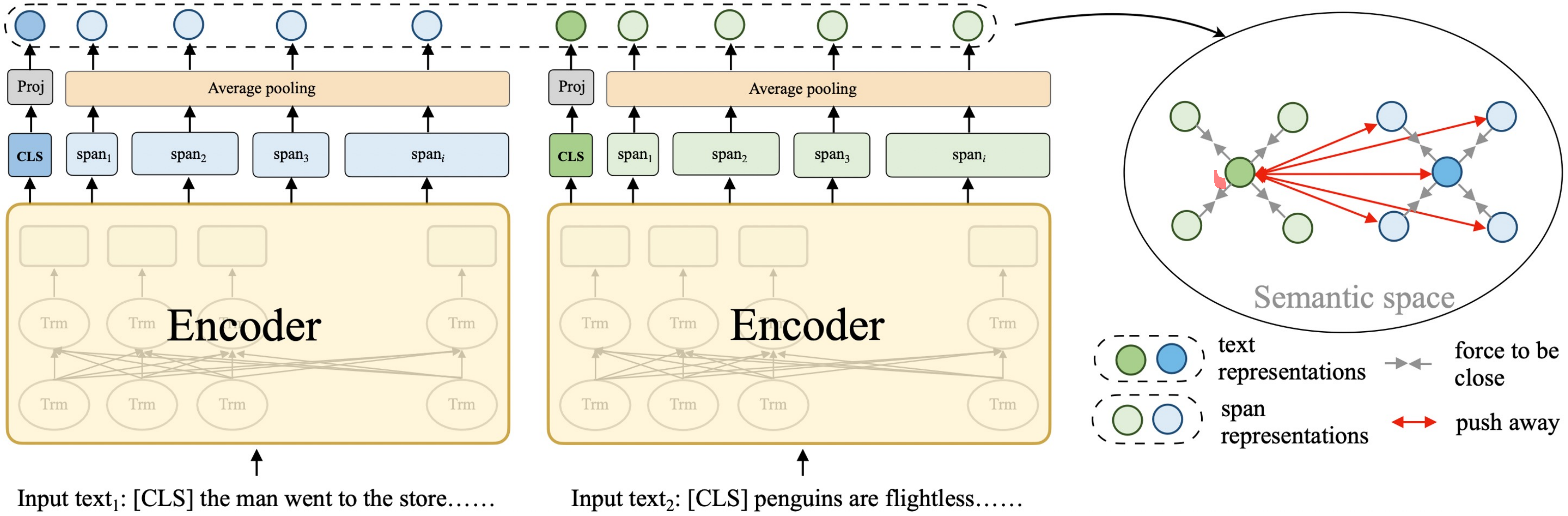
- Leveraging the merits of contrastive learning and autoencoder




(a) Contrastive Learning (b) Auto-encoder (c) Our approach

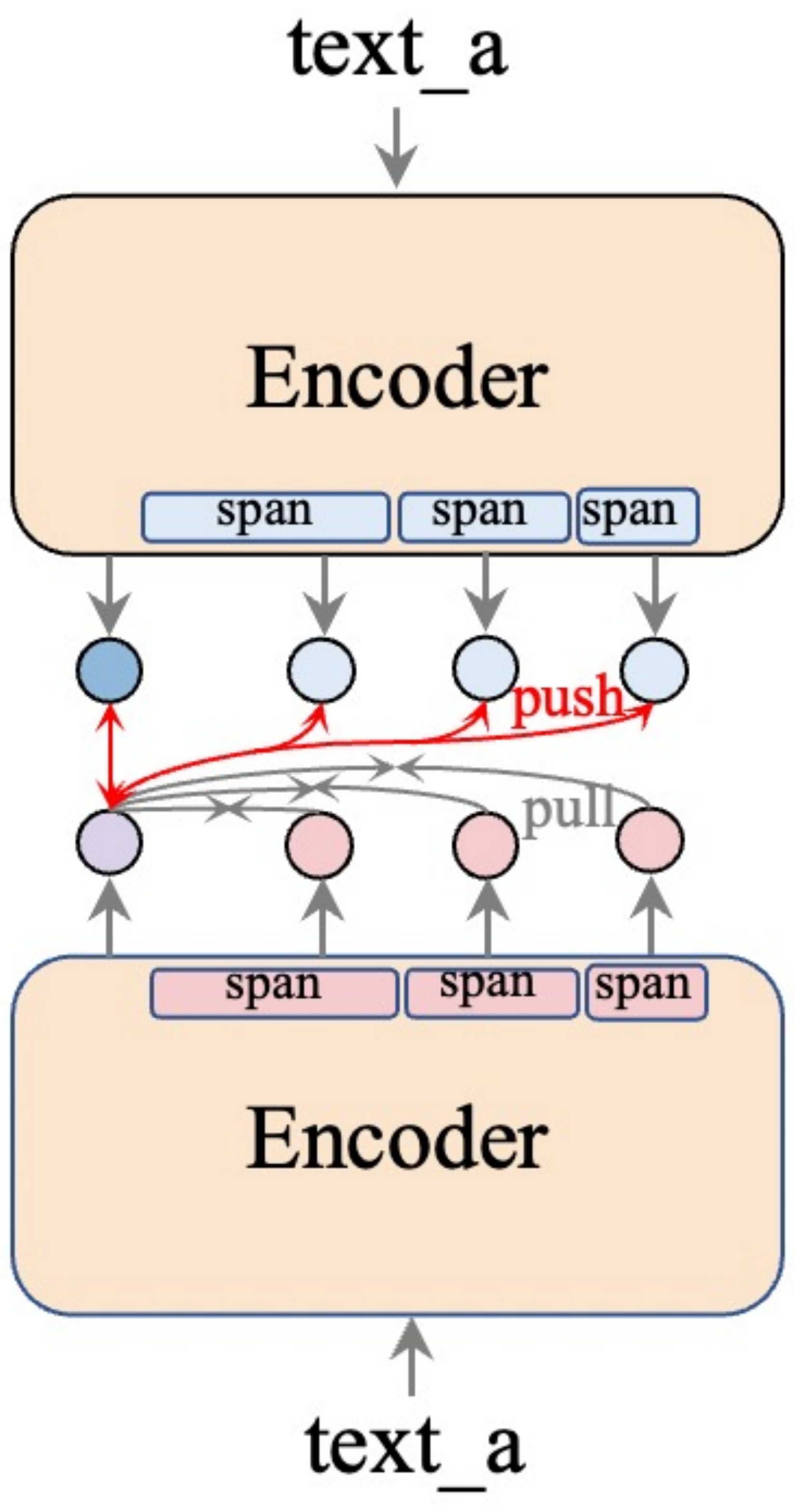
# COSTA

- Key idea: Learning the text sequence representation from its spans via a group-wise contrastive loss




Contrastive span prediction task

# COSTA



(c) Our approach

Improvements:

- learn document-level representations by ”*reconstructing*” its own multiple spans with different granularities
- Only use the encoder

Advantages:

- Learn **discriminative** representations while avoid designing complicated data augmentation techniques
- Retain the **bottleneck** ability while avoid the bypass effect thoroughly
- **Resemble the relevance relationship** between query and the document

# COSTA

## Step 1: Multi-granularity Span Sampling

① Sampling Span length from Beta distribution[1]

$$p_{span} \sim Beta(\alpha, \beta),$$

$$\ell_{span} = p_{span} * (\ell_{max} - \ell_{min}) + \ell_{min},$$

② Sample start position randomly

$$start \sim U(1, n - \ell_{span}).$$

$$end = start + \ell_{span},$$

$$span = [x_{start}, \ldots, x_{end-1}].$$

| | Length |
|---|---|
| Word-level | Whole word |
| Phrase-level | 4-16 |
| Sentence-level | 16-64 |
| Passage-level | 64-128 |

## Step 2: Text Encoding

## Step 3: Group-wise Contrastive Learning

*[1] DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations, ACL 2021*

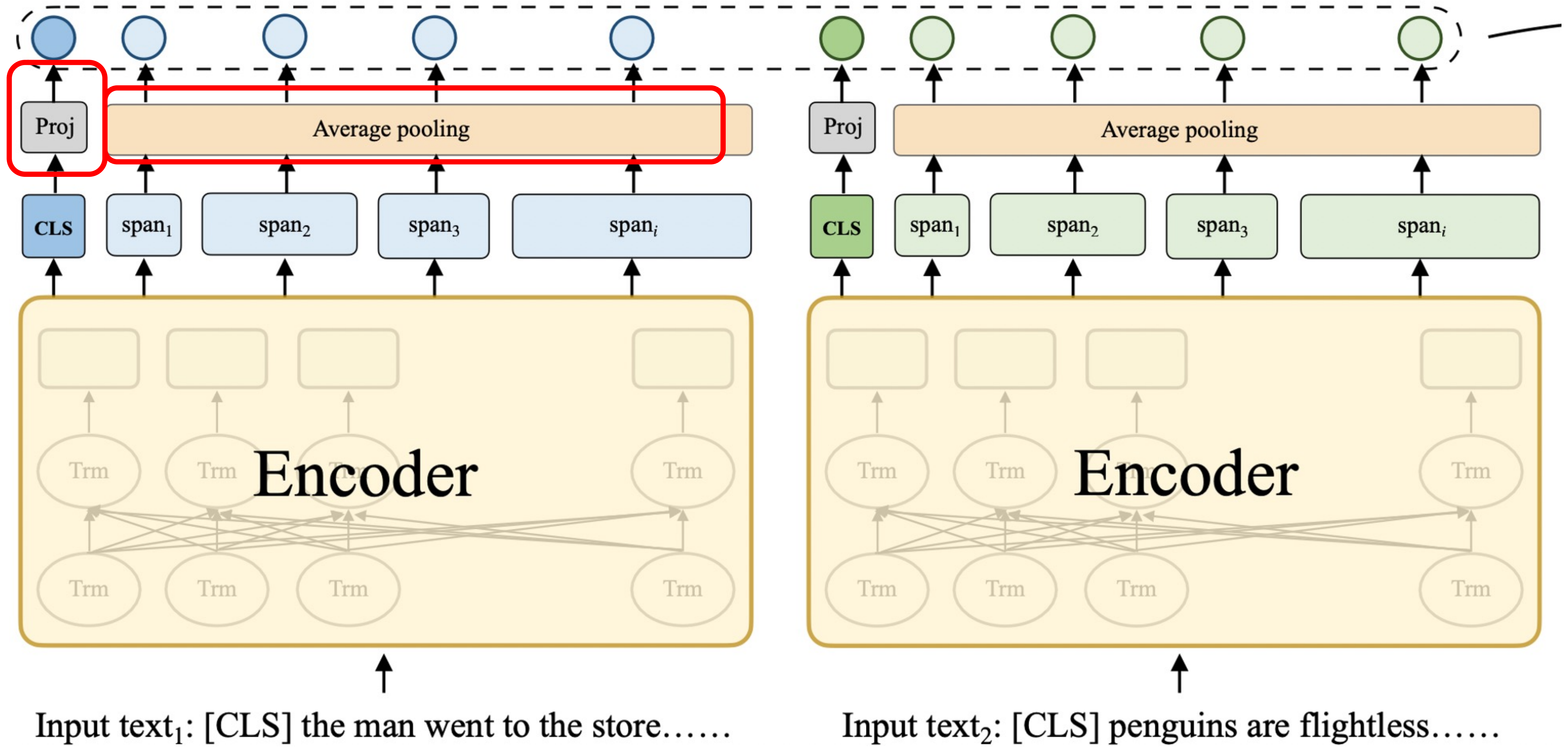
# COSTA

Step 1: Multi-granularity Span Sampling

Step 2: Text Encoding

① Use the [CLS] vector represent the whole sequence

② Use mean-pooling to obtain the span representation




Step 3: Group-wise Contrastive Learning

# COSTA

Step 1: Multi-granularity Span Sampling

Step 2: Text Encoding

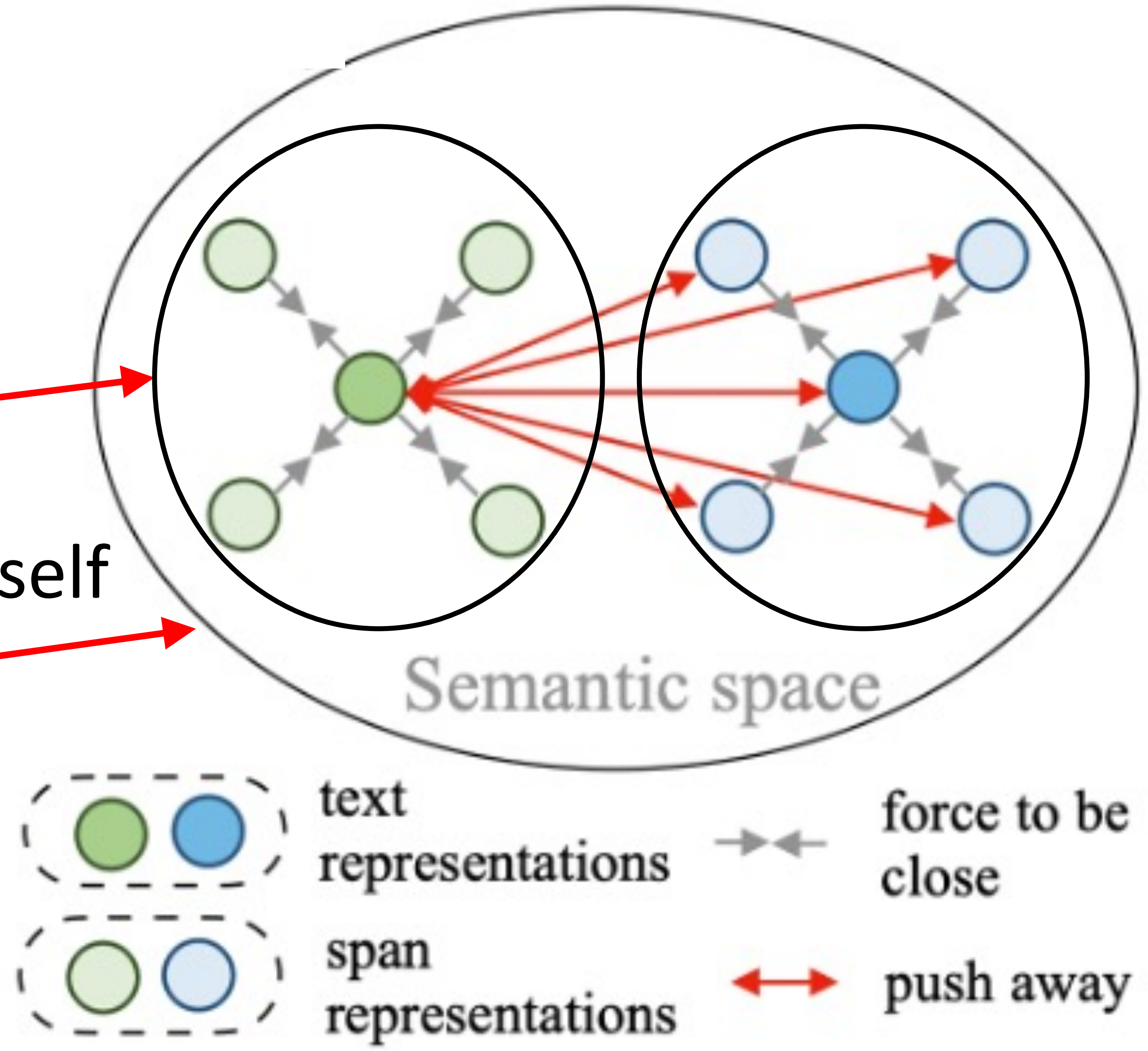
Step 3: Group-wise Contrastive Learning

Positive pairs from a group

$$\mathcal{L}_{GWC} = \sum_{i=1}^{N} -\frac{1}{4T} \sum_{\boxed{p \in S(i)}} \log \frac{\exp(sim\boxed{(\mathbf{z}_i, \mathbf{z}_p)}/\tau)}{\boxed{\sum_{j=1}^{N*(4T+1)} \mathbb{1}_{[i \neq j]} \exp(sim(\mathbf{z}_i, \mathbf{z}_j)/\tau)}}, \quad (8)$$

Except itself

group: document representation and its own spans



# COSTA

- MLM task to learn good span representation

$$\mathcal{L}_{MLM} = -\sum_{\hat{x} \in X} \log p(\hat{x} | X_{\backslash \hat{x}})$$

- Contrastive span prediction task to learn discriminative sequence representations

$$\mathcal{L}_{GWC} = \sum_{i=1}^{N} -\frac{1}{4T} \sum_{p \in S(i)} log \frac{exp(sim(\mathbf{z}_i, \mathbf{z}_p)/\tau)}{\sum_{j=1}^{N*(4T+1)} \mathbb{1}_{[i \neq j]} exp(sim(\mathbf{z}_i, \mathbf{z}_j)/\tau)}, \tag{8}$$

- Final loss:

$$\mathcal{L}_{total} = \lambda \mathcal{L}_{GWC} + \mathcal{L}_{MLM},$$

# Experiment Setting

- Pretraining datasets：
  - Wikipedia,  over 10 million documents
- 4 large-scale downstream dense retrieval tasks：
  - MS MARCO Document ranking and TREC DL Document ranking
  - MS MARCO Passage ranking and TREC DL Passage ranking
- Baseline models:
  - BM25, BERT, PROP, B-PROP, ICT, SEED

# Main Results

| Model | MARCO Dev Passage | | TREC2019 Passage | |
|---|---|---|---|---|
| | MRR@10 | R@1000 | NDCG@10 | R@1000 |
| *Sparse retrieval models* | | | | |
| BM25 | 0.187 | 0.857 | 0.501 | 0.745 |
| DeepCT[6] | 0.243 | 0.905 | 0.551 | - |
| Best TREC Trad[5] | - | - | 0.554 | - |
| *Fine-tuning with official BM25 negatives* | | | | |
| BERT | 0.316 | 0.941 | 0.616 | 0.704 |
| ICT | 0.324 | 0.938 | 0.618 | 0.705 |
| PROP | 0.320 | 0.948 | 0.586 | 0.709 |
| B-PROP | 0.321 | 0.945 | 0.603 | 0.705 |
| SEED[29] | 0.329 | 0.953 | - | - |
| SEED(ours) | 0.331* | 0.950* | 0.625* | 0.733*† |
| COSTA | **0.342***†‡ | **0.959***† | **0.635***†‡ | **0.773***†‡ |
| *Fine-tuning with static hard negatives* | | | | |
| BERT | 0.335 | 0.957 | 0.661 | 0.769 |
| ICT | 0.339 | 0.955 | 0.670 | 0.775 |
| PROP | 0.337 | 0.951 | 0.673 | 0.771 |
| B-PROP | 0.339 | 0.952 | 0.672 | 0.774 |
| SEED | 0.342* | 0.963 | 0.679* | 0.782*† |
| COSTA | **0.366***†‡ | **0.971***† | **0.704***†‡ | **0.816***†‡ |

| Model | MARCO Dev Doc | | TREC2019 Doc | |
|---|---|---|---|---|
| | MRR@100 | R@100 | NDCG@10 | R@100 |
| *Sparse retrieval models* | | | | |
| BM25 | 0.277 | 0.808 | 0.519 | 0.395 |
| DeepCT[6] | 0.320 | - | 0.544 | - |
| Best TREC Trad[5] | - | - | 0.549 | - |
| *1st iteration: Fine-tuning with static hard negatives* | | | | |
| BERT | 0.358 | 0.869 | 0.563 | 0.266 |
| ICT | 0.364 | 0.873 | 0.566 | 0.273 |
| PROP | 0.361 | 0.871 | 0.565 | 0.269 |
| B-PROP | 0.365 | 0.871 | 0.567 | 0.268 |
| SEED | 0.372* | 0.879* | 0.573* | 0.272 |
| COSTA | **0.395***†‡ | **0.894***†‡ | **0.582***†‡ | **0.278*** |
| *2nd iteration: Fine-tuning with static hard negatives* | | | | |
| BERT | 0.389 | 0.877 | 0.594 | 0.301 |
| ICT | 0.396 | 0.882 | 0.605 | 0.303 |
| PROP | 0.394 | 0.884 | 0.596 | 0.298 |
| B-PROP | 0.395 | 0.883 | 0.601 | 0.305 |
| SEED | 0.396 | 0.902* | 0.605* | 0.307 |
| COSTA | **0.422***†‡ | **0.919***†‡ | **0.626***†‡ | **0.320***†‡ |

- Beat the baselines significantly!

# Comparison with Different Fine-tuning Strategies

**Table 3: Comparison between COSTA and advanced dense retrieval models using complicated fine-tuning strategies on the MARCO Dev Passage. Best results are marked bold.**

| Model | MRR@10 | R@1000 |
|---|---|---|
| ANCE[40] | 0.330 | 0.959 |
| TCT-ColBERT[27] | 0.335 | 0.964 |
| TAS-B[18] | 0.343 | **0.976** |
| ADORE+STAR[40] | 0.347 | - |
| RoctetQA w/o Data Aug [33] | 0.364 | - |
| COSTA | **0.366** | 0.971 |

Training Technologies

- **In-batch negative**
- **Static Hard negative mining**
- Dynamic Hard Negative (ANCE, ADORE)
- Data Augmentation (Rocket QA)
- Distillation (TCT-ColBERT, TAS)
- Denoising False Negatives (RocketQA)

- Fine-tuning with simple strategies COSTA performs better than these advanced dense retrieval models with complicated fine-tuning strategies

# Breakdown Analysis

- The impact of span type and span number

Table 4: The performance of COSTA with different span granularities. Best results are marked bold.

| Method | MRR@10 | R@1000 |
|---|---|---|
| Base | **0.335** | **0.952** |
| w/o word-level | 0.334 | 0.952 |
| w/o phrase-level | 0.331 | 0.953 |
| w/o sentence-level | 0.331 | 0.947 |
| w/o paragraph-level | 0.326 | 0.940 |

Table 5: Performance comparison of COSTA with different span numbers. Best results are marked bold.

| Span Number | 3 | 5 | 10 | 20 |
|---|---|---|---|---|
| MRR@10 | 0.335 | **0.339** | 0.332 | 0.320 |
| R@1000 | 0.952 | **0.953** | 0.949 | 0.946 |

- Longer spans are most useful than short spans
- Neither too many spans nor too little spans for a text

# The discriminative ability of COSTA

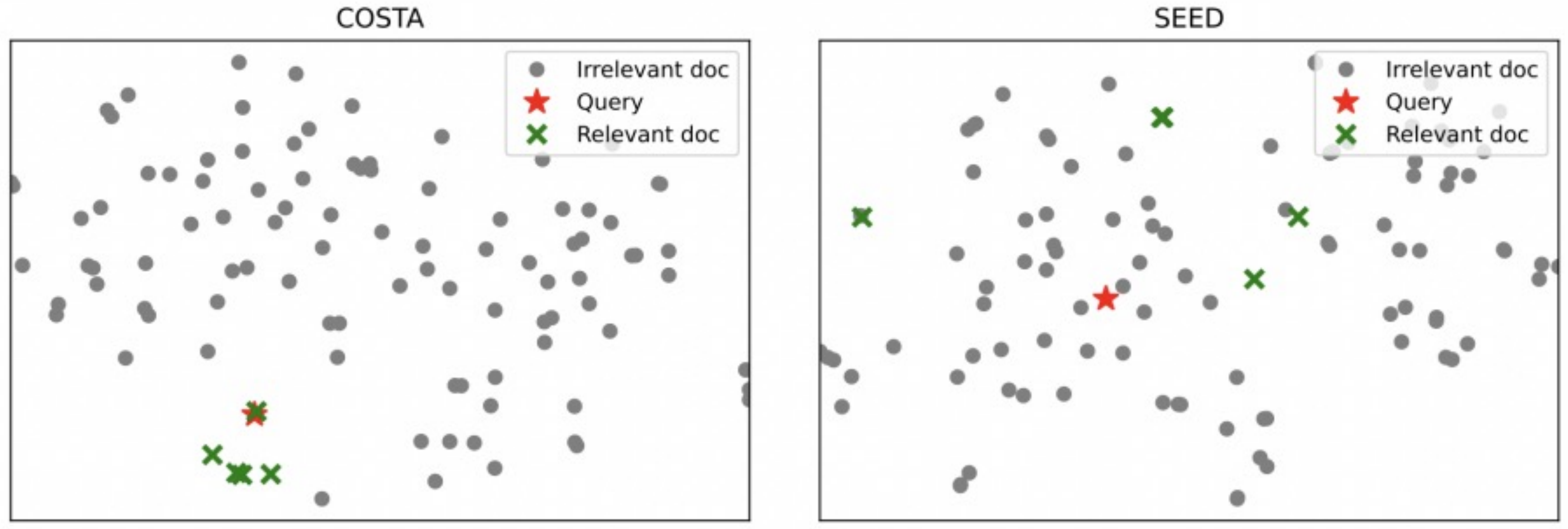



**Figure 3: The t-SNE plot of query and document representations for SEED and COSTA. The QID is 47923 and is from TREC2019 Passage test set.**

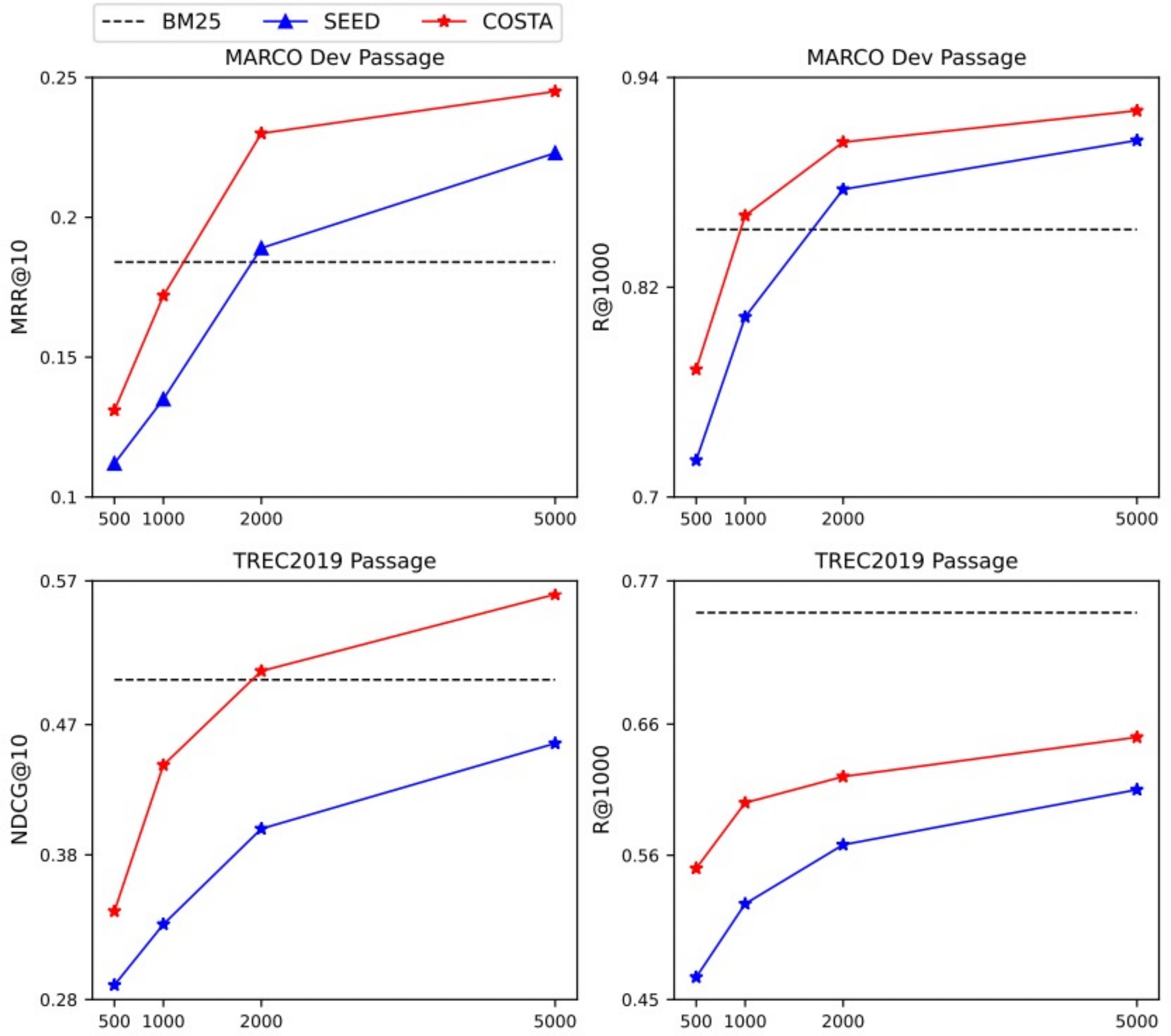



**Figure 4: Fine-tuning with limited supervised data. The x-axis indicates the number of training queries.**

- The representations produced by COSTA are more discriminative than from SEED

# Conclusion

- We proposed a novel contrastive span prediction task to pre-train a discriminative text encoder for dense retrieval.
- COSTA can leverage the merits of both the autoencoder-based language models and contrastive learning to produce high-quality representations.
- COSTA outperforms several strong baselines and can produce discriminative representations for dense retrieval verified by visualization analysis and the low-resource setting

# Future work

- Simple yet effective data augmentations for information retrieval?
- What contributes to the relevance matching?
- Larger model, more data lead to **strong zero-shot performance?**
- **Prompt for ranking?**

Code is released at https://github.com/Albert-Ma/COSTA

## Fine-tuning Results

| MS MARCO Passage Retrieval | MRR@10 | Recall@1000 | Files |
|---|---|---|---|
| COSTA (BM25 negs) | 0.342 | 0.959 | Model, Dev(MARCO format), Dev (TREC format) |
| COSTA (hard negs) | 0.366 | 0.971 | Model, Dev (MARCO format), Dev (TREC format) |

| TREC 2019 Passage Retrieval | NDCG@10 | Recall@1000 | Files |
|---|---|---|---|
| COSTA (BM25 negs) | 0.635 | 0.773 | Model, Test (TREC format) |
| COSTA (hard negs) | 0.704 | 0.816 | Model, Test (TREC format) |

Run the following code to evaluate COSTA on MS MARCO Passage dataset.

```
./eval/eval_msmarco_passage.sh   ./marco_pas/qrels.dev.tsv ./costa_hd_neg8_e2_bs8_fp16_mrr10_366_r10
```

You will get

```
#####################
MRR @ 10: 0.36564396006731276
QueriesRanked: 6980
#####################
```

Run the following code to evaluate COSTA on TREC2019 Passage dataset.

```
./eval/trec_eval -m ndcg_cut.10 -m recall.1000  -c -l 2 ./marco_pas/qrels.dl19-passage.txt ./costa_l
```

You will get

```
recall_1000             all     0.8160
ndcg_cut_10             all     0.7043
```

# Thanks！


**Xinyu Ma**

maxinyu17g@ict.ac.cn