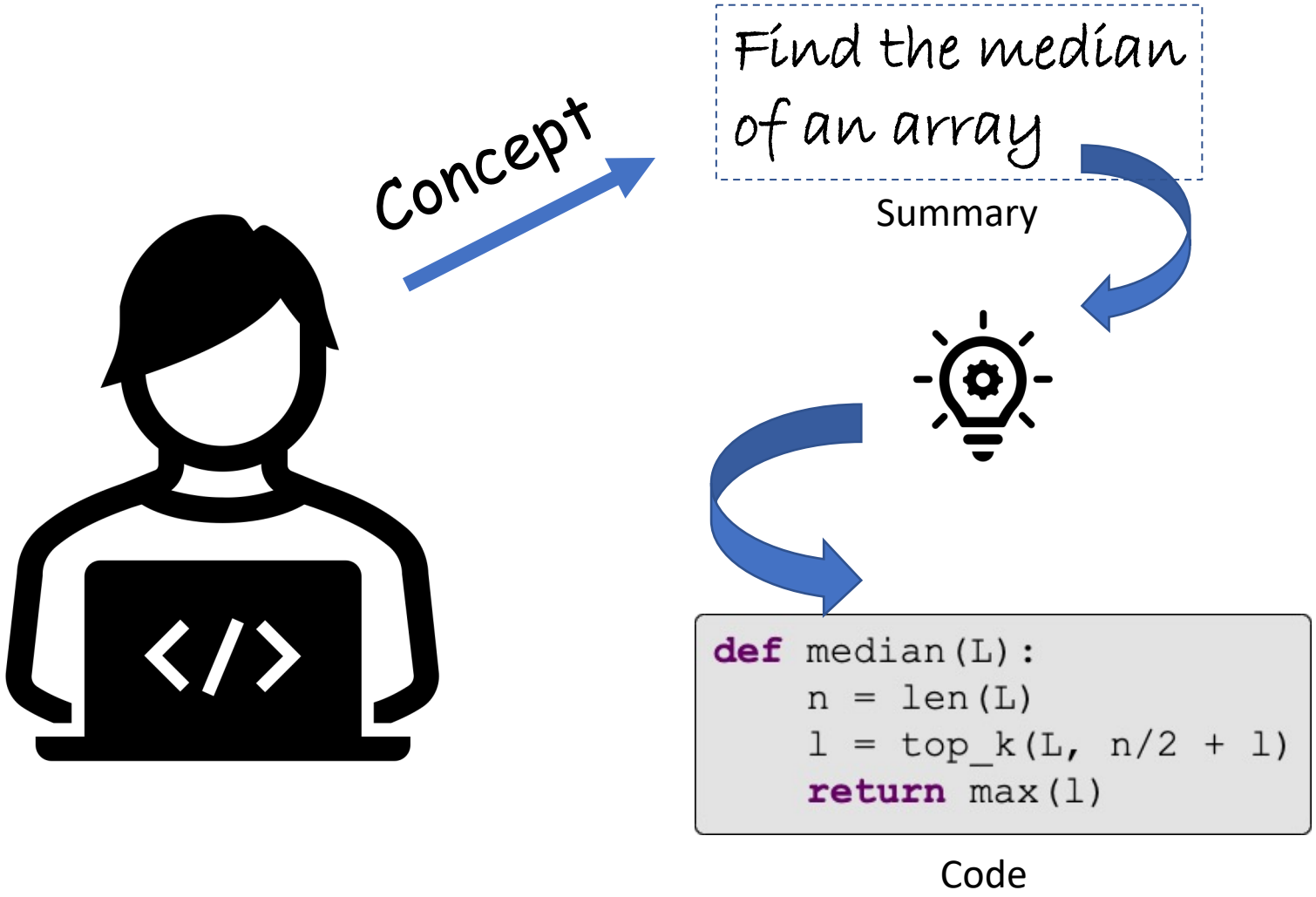
Retrieval Augmented Code Generation and Summarization

Md Rizwan Parvez[§], **Wasi Uddin Ahmad**[§], **Saikat Chakraborty**[†]
Baishakhi Ray[†], **Kai-Wei Chang**[§]
[§]University of California, Los Angeles, [†]Columbia University

EMNLP-Findings 2021

Motivation



Slide idea: Graham Neubig

Motivation

Concept

Sort my_tensor in descending order

Search API guidelines

Python sorted in descending order

Google PyTorch TensorFlow stackoverflow GitHub

Browse thru. top few results

```
torch.sort(input, dim=-1, descending=False, stable=False, *, out=None) ->
(Tensor, LongTensor)
```

Parameters

- **input** (*Tensor*) – the input tensor.
- **dim** (*int, optional*) – the dimension to sort along
- **descending** (*bool, optional*) – controls the sorting order (ascending or descending)

Adapt the results

```
my_tensor.sort(descending=True)
```

Slide idea: Graham Neubig

REDCODER

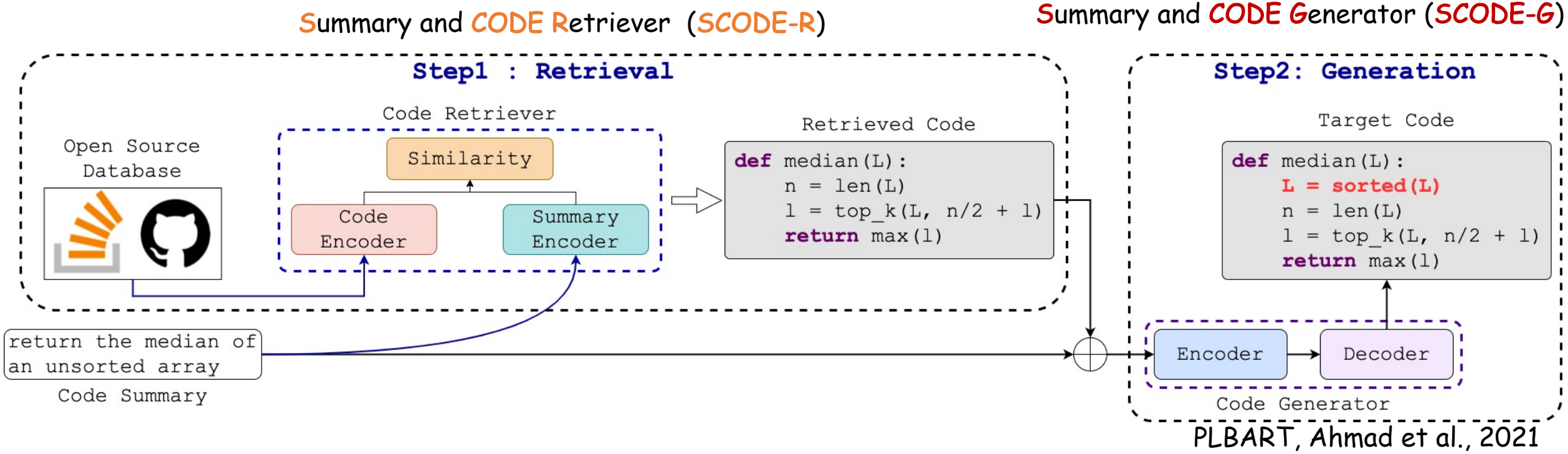
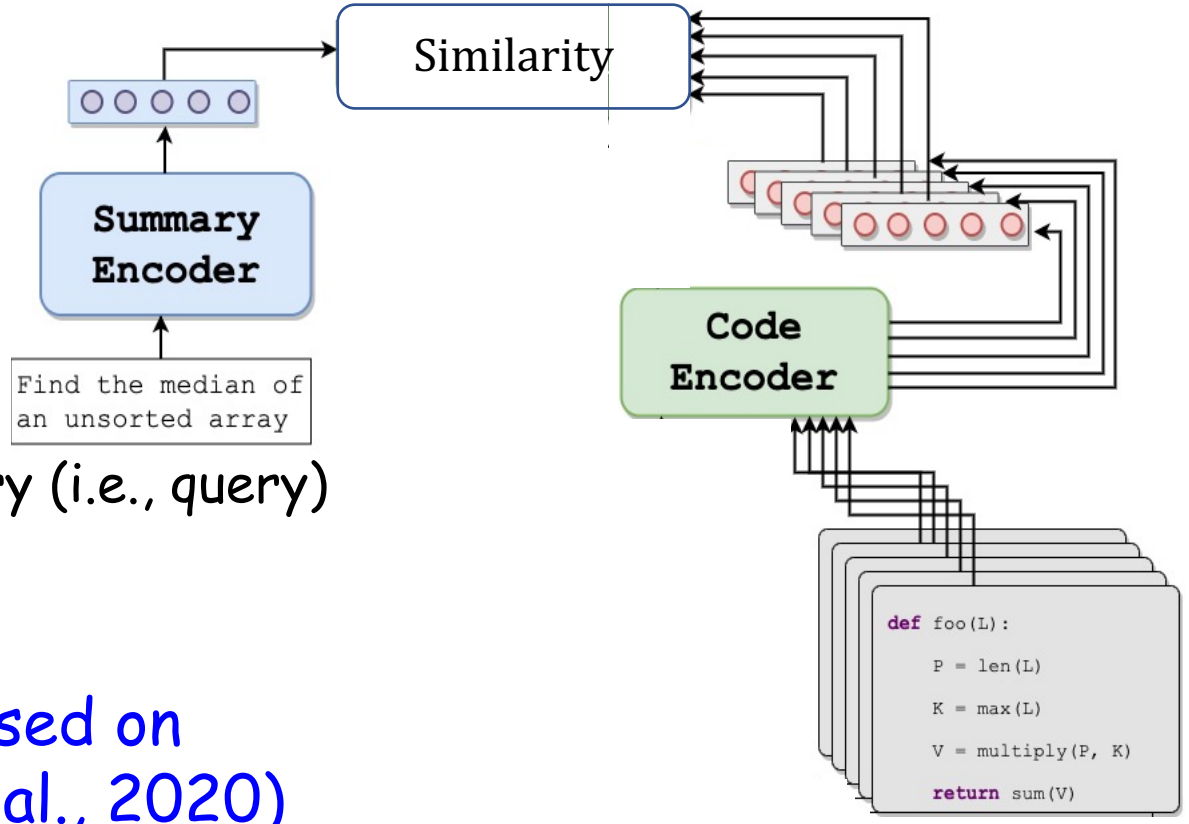


Fig: Retrieval augmentED CODe gEneration and summaRization framework (REDCODER)

Sparse Vs Dense SCODE-R

- ✓ Must be fast
- ✓ Needs understanding of both natural and programming languages



Input summary (i.e., query)

Candidate code (i.e., docs)

SCODE-R is based on DPR (Karpukhin et al., 2020)

SCODE-R Training

- ✓ As a binary classification problem
- ✓ Using the same <summary, code> training set in our final gen/sum task
- ✓ No hard-negatives

Deep - copy an observation dict

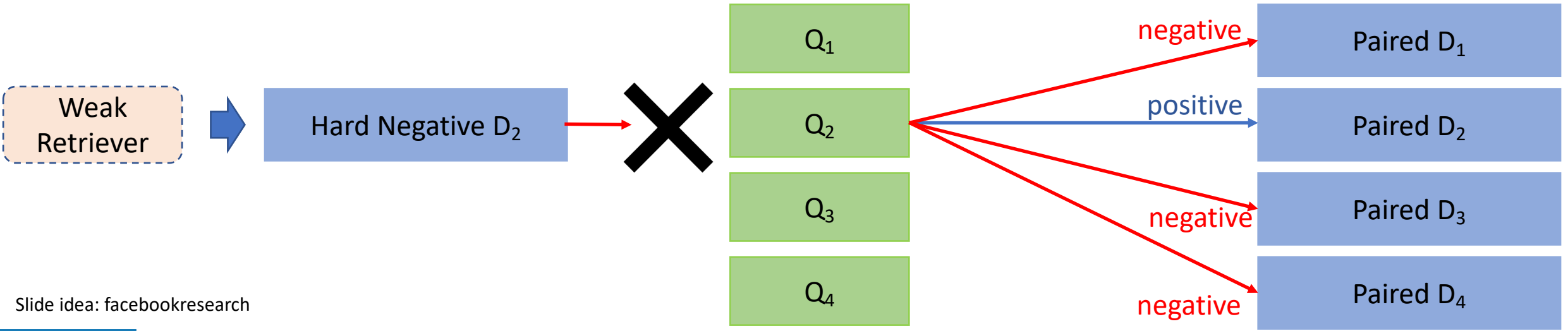
Target

```
def copy(self, deep=True, data=None):
  self.variable.copy(deep=deep,\
                     data=data)
```

Retrieved

```
def copy_obs_dict(obs):
    return {k:np.copy(v) \
            for k,v in obs.items()}
```

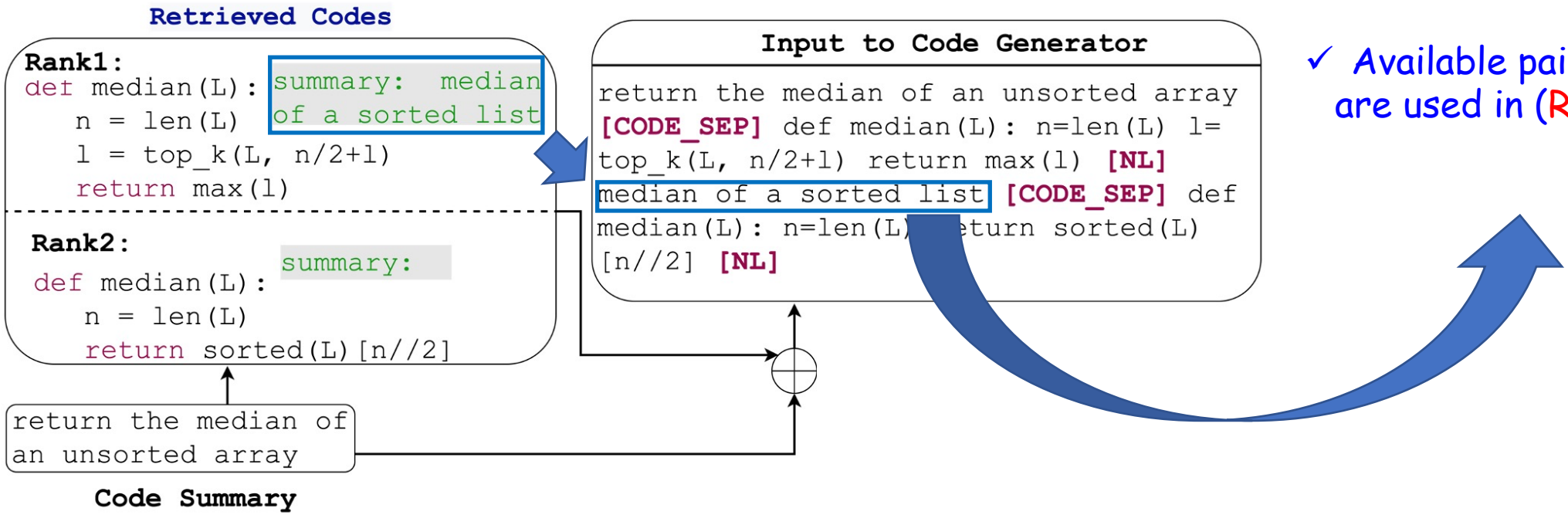
Example: A relevant yet not same retrieved code



Training minibatch

Slide idea: facebookresearch

SCODE-G



- ✓ SCODE-G in REDCODER uses retrieved candidate code only
- ✓ Available paired summaries are used in (REDCODER-ext)

Evaluation Settings

Baselines

Retrieval Based	BM25 SCODE-R
Generative	CodeBERT GraphCodeBERT CodeGPT-adapted PLBART
Retrieval Augmented Generative	BM25 + PLBART REDCODER REDCODER-EXT

Benchmark

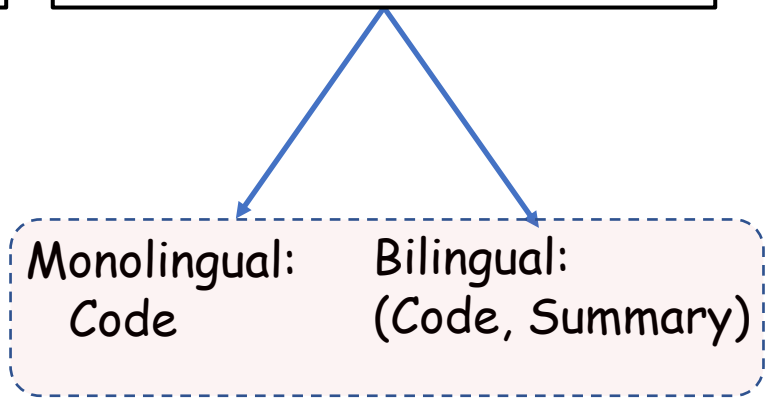
CodeXGlue: Lu et al. (2021)

Metrics

- ✓ BLEU
- ✓ CodeBLEU
- ✓ EM

Retrieval DB

CSNET: Husain et al. (2019)



By default, target output is removed

Evaluation

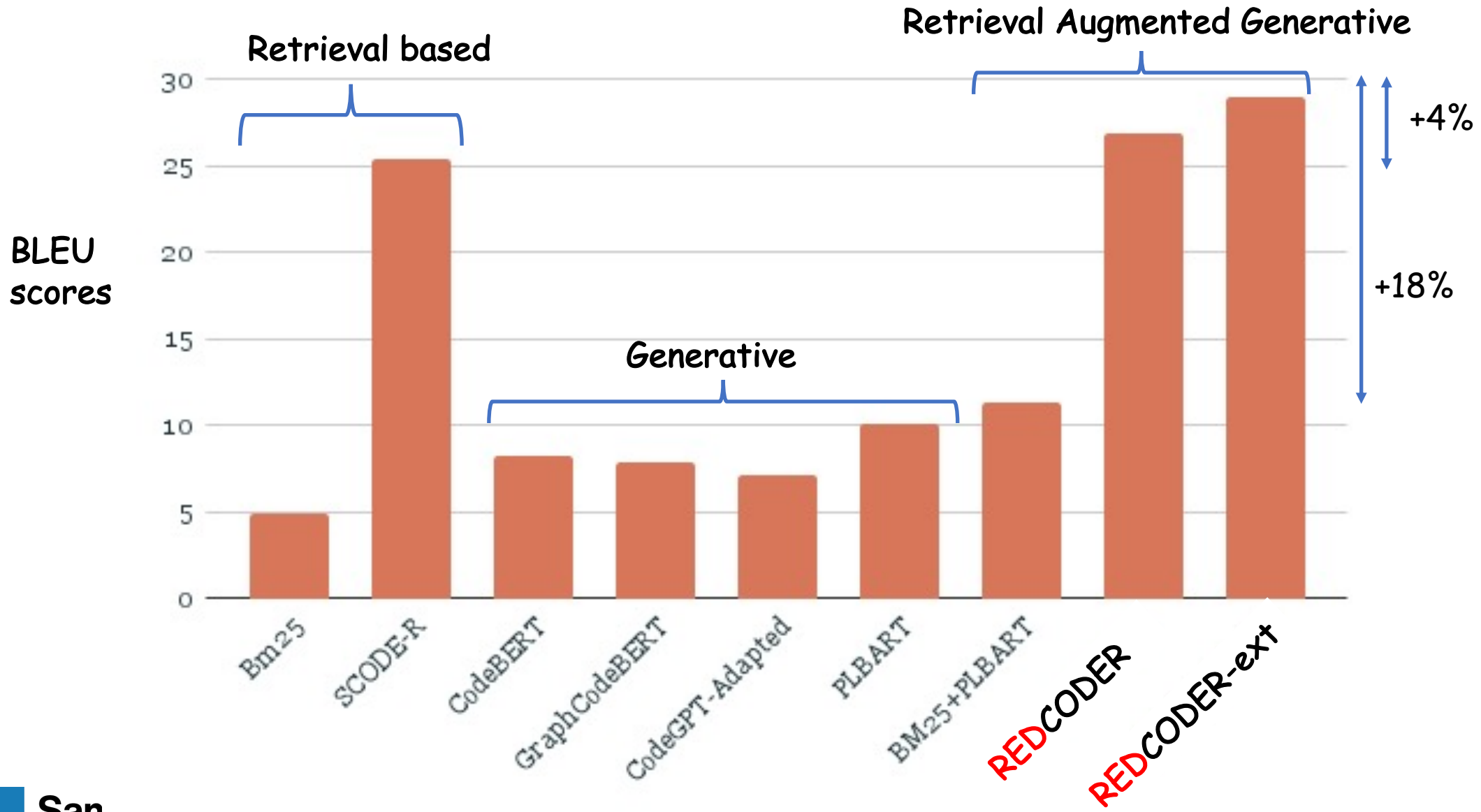


Table: Code gen. performances

Qualitative Example

Reference (Gold Output)

Input summary: Get the MuxerStream at the given position.

```
1 public MuxerStream getMuxer (int streamIndex)
2   throws java.lang.InterruptedException, java.io.IOException {
3   long cPtr = VideoJNI.MuxerStream_getMuxer(swigCPtr, this, streamIndex);
4   return (cPtr == 0) ? null : new MuxerStream (cPtr, false);
5 }
```

PLBART Prediction **[BLEU: 0.1439]**

```
1 public MuxerStream getMuxerStream (int position) {
2   if (muxerStream == null) {
3     muxerStream = new MuxerStream (this, position)
4   }
5   return muxerStream;
6 }
```

PLBART fails to predict the diverse identifiers (in red color) whereas REDCODER succeeds

Redcoder-ext Prediction **BLEU: 80.6**

```
1 public MuxerStream getMuxer (int streamIndex)
2   throws java.lang.InterruptedException, java.io.IOException {
3   long cPtr = VideoJNI.MuxerStream_getMuxer(swigCPtr, this, streamIndex);
4   return (cPtr == 0) ? null : new MuxerStream (cPtr, false);
5 }
```

Thank You!

Questions?

https://github.com/rizwan09/REDCODER