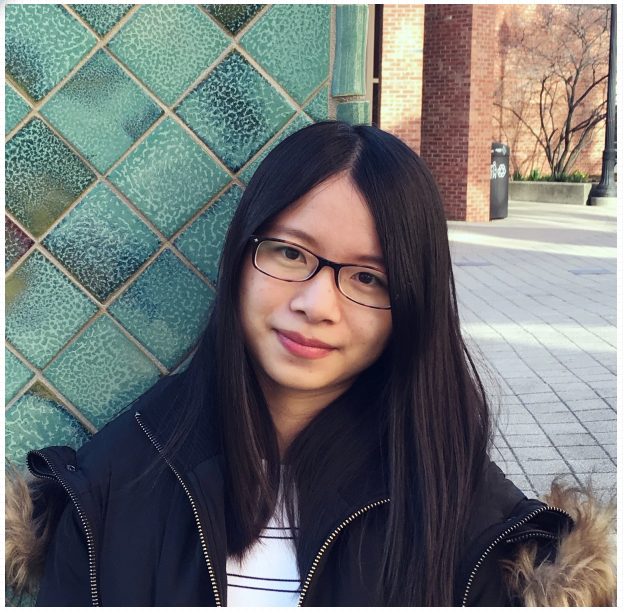
Coacor: Code Annotation for Code Retrieval with Reinforcement Learning

International World Wide Web Conference 2019
CCF A-class Conference

The Ohio State University

Authors



Ziyu Yao
Ph.D. student

Jayavardhan Reddy Peddamail
Master student

Huan Sun
Assistant professor
Since 2016

Department of Computer Science and Engineering
The Ohio State University

CONTENTS

01 Motivation

01 Motivation

*Previous code annotation work focused on getting a large n-gram overlap **between generated and human-provided annotations.***

Example 2 from StaQC-test set	
SQL Code	SELECT Group_concat(DISTINCT(p.products_id)) AS comma_separated, COUNT(DISTINCT p.products_id) AS product_count FROM ...
Human-provided	how to count how many comma separated values in a group_concat
RL^{MRR}	group_concat count concatenate distinct comma group mysql concat column in one row rows select multiple columns of same id result

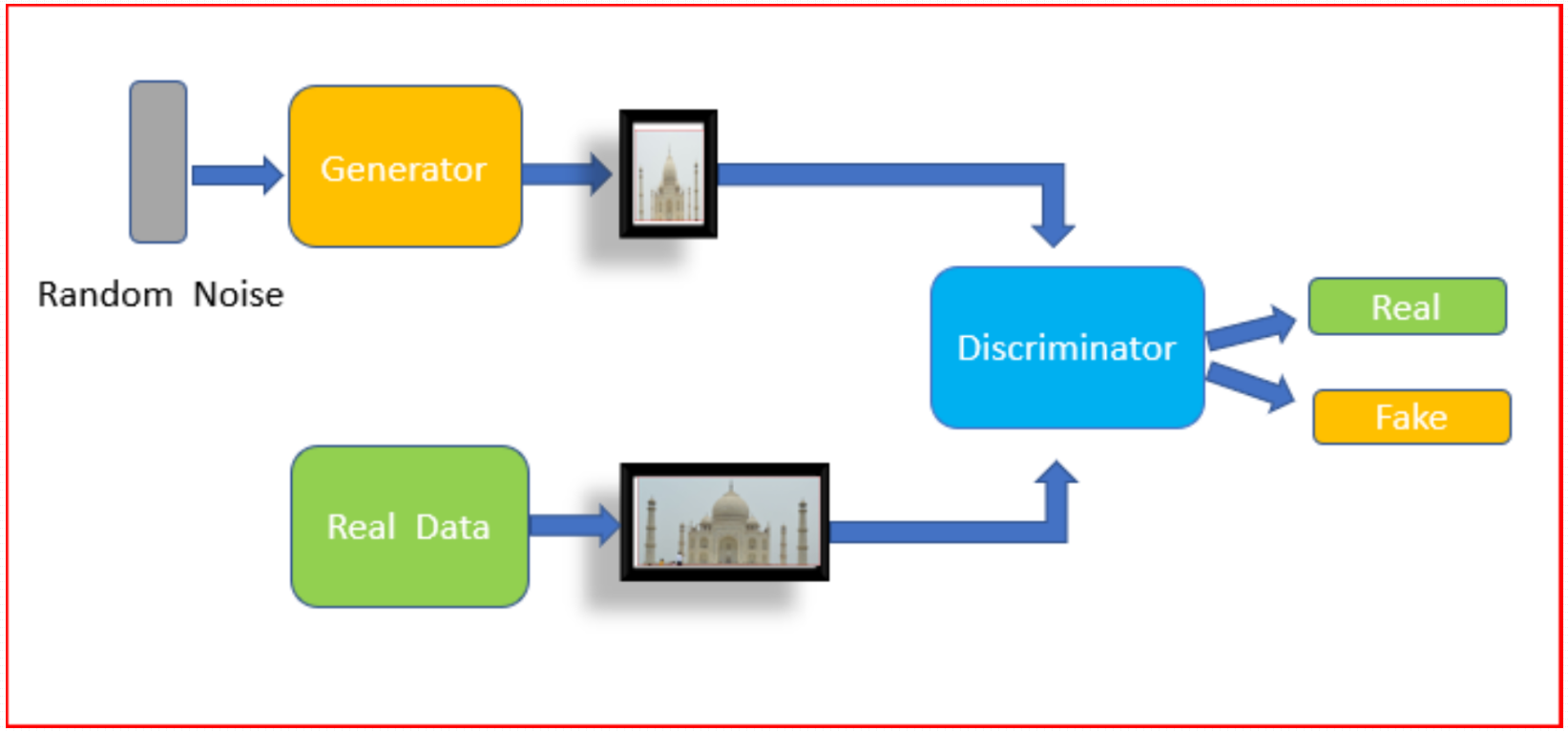
→ ***How useful?***

→ ***For code retrieval !***

*We are the first to examine **the real usefulness** of the generated code annotations and **how they can help a relevant task**, i.e., code retrieval.*

01 Motivation

Machine-machine collaboration mechanism: Generative Adversarial Network (GAN)



Code Annotation (CA): second view of a code snippet to assist code retrieval !

If the CA model is useful, then demonstrate that it can produce richer and more detailed annotations.

01 Motivation

Contributions:

1. Explore a novel perspective of generating useful code annotations for code retrieval. Do not emphasize the n-gram overlap between the generated annotation and the human-provided one. Examine the real usefulness !

2. Develop an effective RL-based framework with a novel rewarding mechanism, in which a code retrieval model is directly used to formulate rewards and guide the annotation generation.

3. Conduct extensive experiments by comparing coacor with various STOA baselines. Show significant improvements of code retrieval performance on both a widely used benchmark dataset and a recently collected large-scale dataset.

02 Background

02 Background

Code Retrieval (CR): Given an NL Query Q, a model F_r will be learnt to retrieve the highest scoring code snippet $C^* \in \mathbb{C}$.

$$C^* = \underset{C \in \mathbb{C}}{\operatorname{argmax}} F_r(Q, C) \tag{1}$$

Code Annotation (CA): For a given code snippet C, the goal is to generate an NL annotation N^* that maximizes a scoring function F_a:

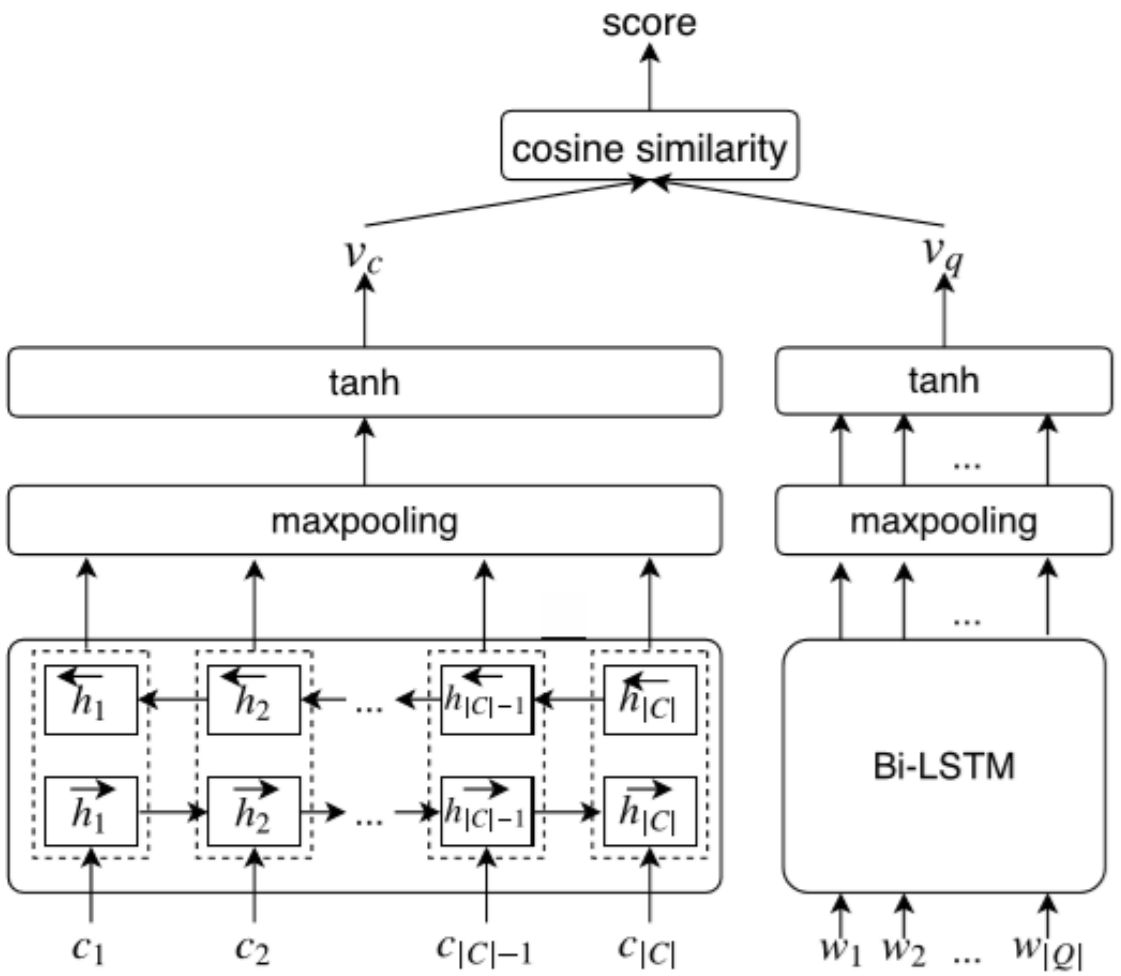
$$N^* = \underset{N}{\operatorname{argmax}} F_a(C, N) \tag{2}$$

Combined for Code Retrieval !

QC-based CR Model: Query Code-based Code Retrieval Model

QN-based CR Model: Query Annotation-based Code Retrieval Model

02 Background: Base Code Retrieval Model



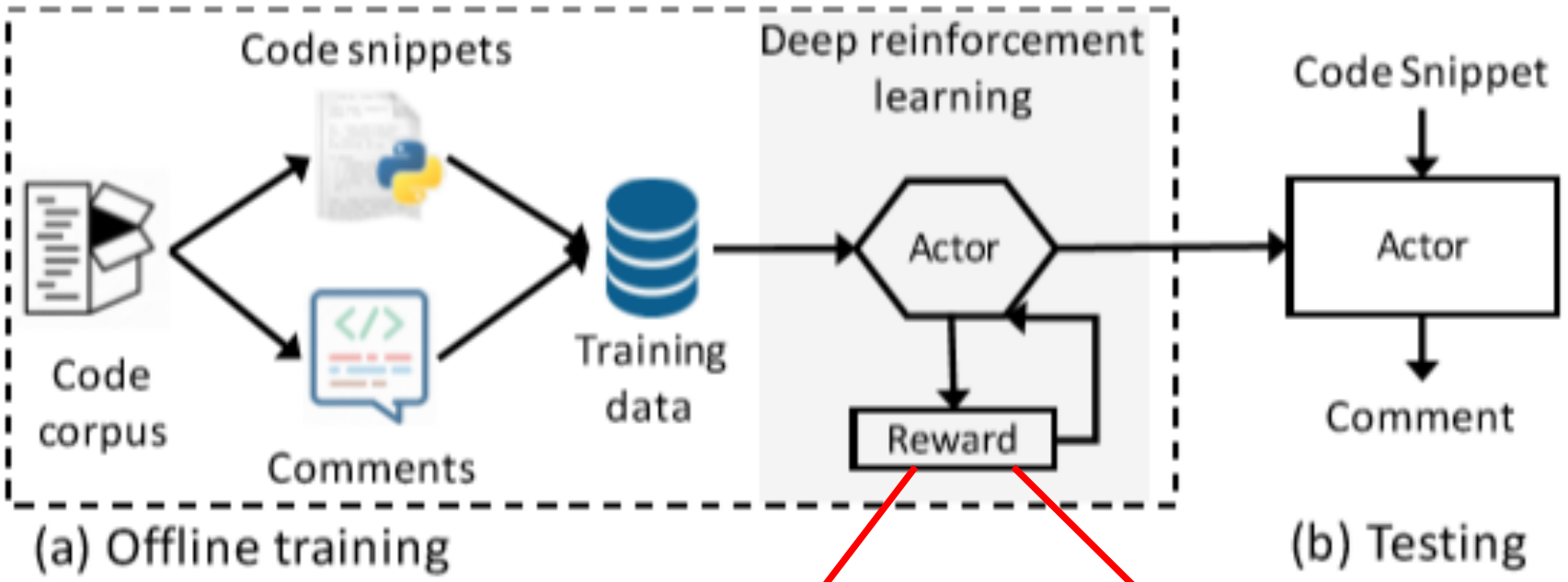
1.Code Snippet

or

2.Code Annotation

Natural Language Query

02 Background: Deep Reinforcement Learning for Code Annotation



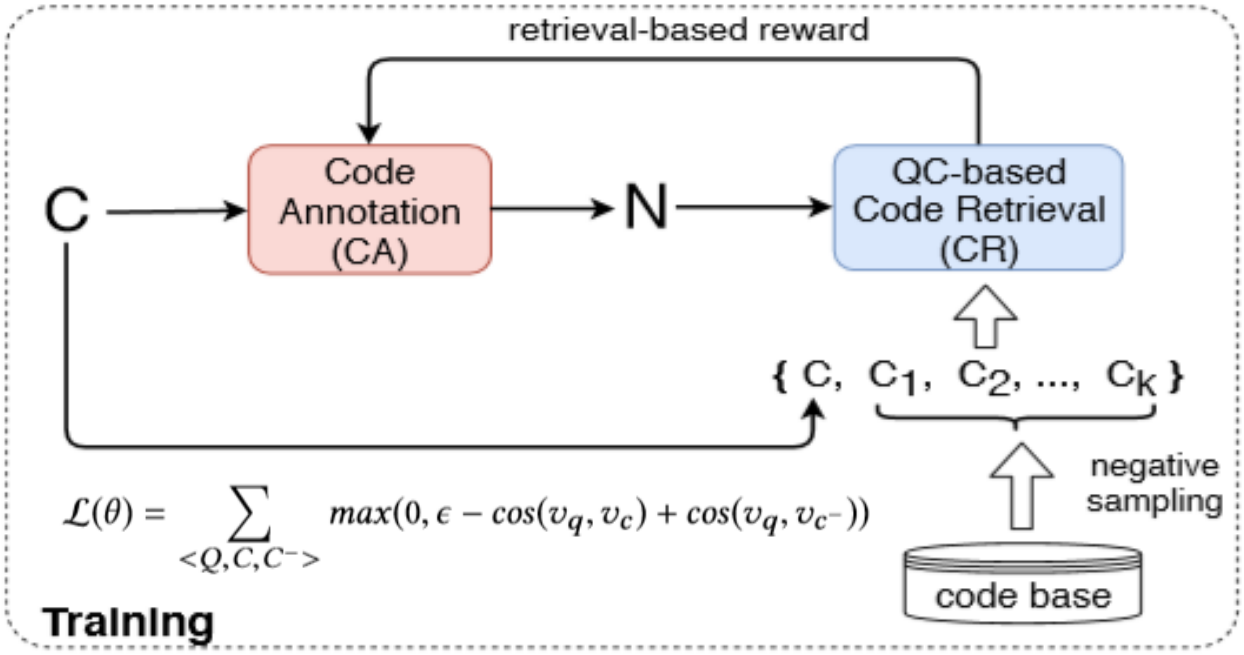
Previous: the n-gram overlap between the generated annotation and the human-provided one / BLEU score

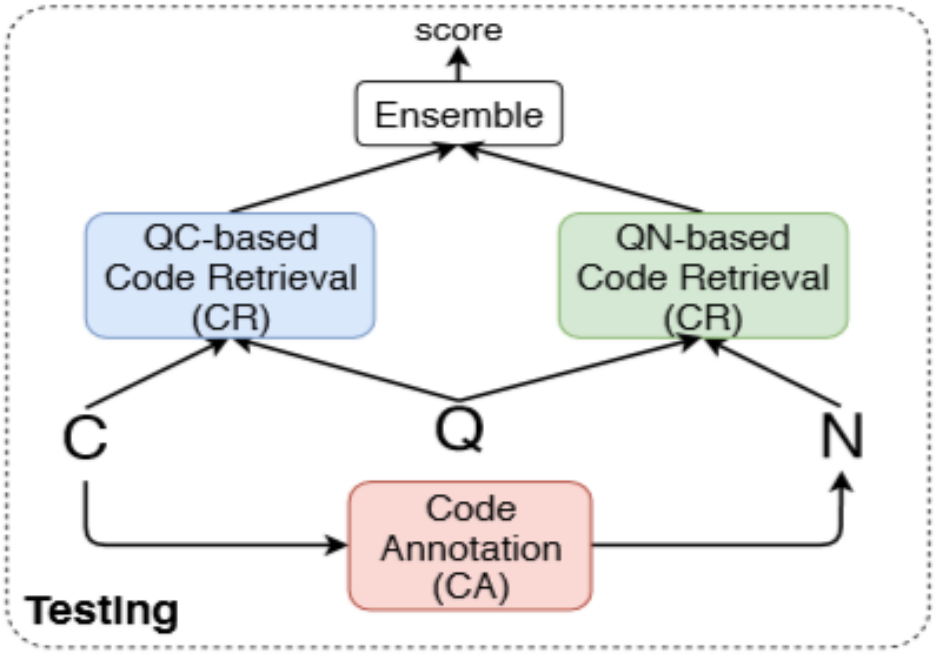
Now: the performance on a base code retrieval model !

Wan Y, Zhao Z, Yang M, et al. Improving automatic source code summarization via deep reinforcement learning[C]// Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering. ACM, 2018: 397-407.

03 Framework

03 Framework of Coacor





Code snippet C:

```
SELECT id, date, site, url FROM links
WHERE publish = "yes" AND date = (
   SELECT date FROM links
   WHERE date < '2014/02/25'
   ORDER BY date DESC LIMIT 1 )
AND category!= 'Adult'
ORDER BY date DESC, clicks DESC
LIMIT 200
```

NL queries Qs for C:

(1) "fetch past records closest to given date and sort"

(2) "select 200 most popular non-adult links with date earlier than 2014/02/25"

(3) "mysql and php: select all fields from the same date"

Generated code annotation N:

Method	Generated N
MLE	mysql query to get the last record from a table
RL^{BLEU}	how to select date in mysql sql query
RL^{MRR}	mysql order last date where day select maximum datetime column in multiple dates sqlite sql limit query result specific timestamp

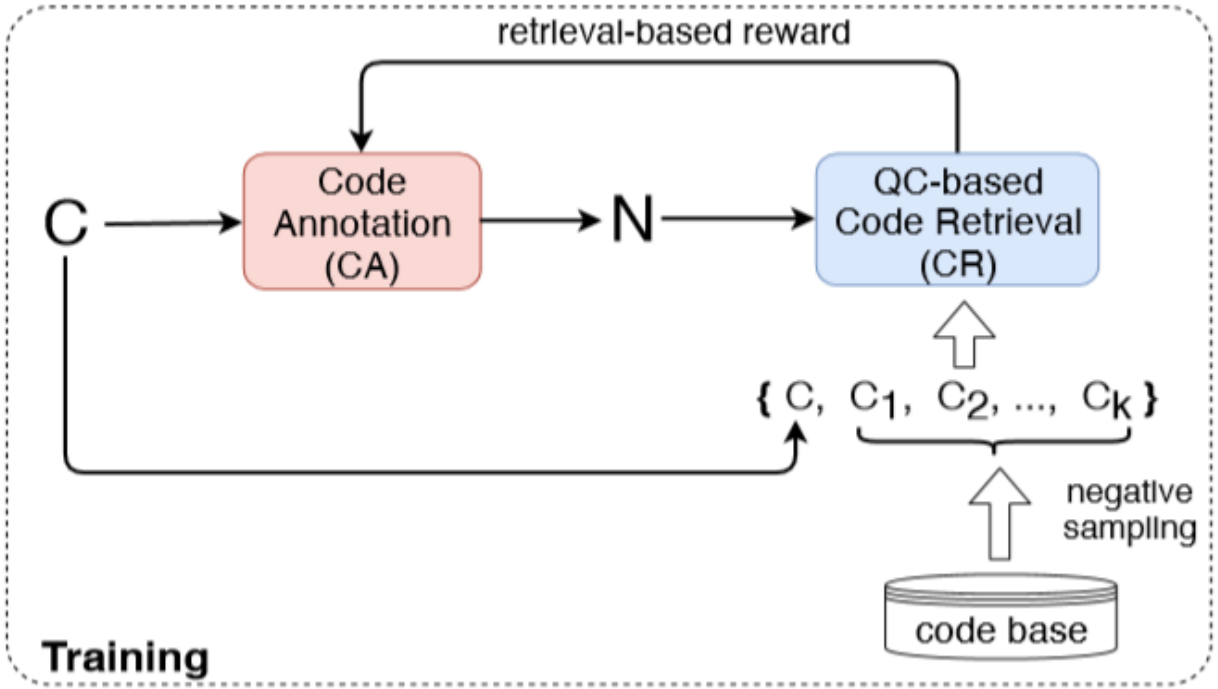
Example

03 Training Procedure of Coacor

Algorithm 1 : Training Procedure for CoaCor.

Input: <NL query, code snippet> (QC) pairs in training set, number of iterations E.

1: Train a base code retrieval model based on QC pairs, according to Eqn. (5).
2: Initialize a base code annotation model (ϕ) and pretrain it via MLE according to Eqn. (7), using Q as the desired N for C.
3: Pretrain a critic network (ρ) according to Eqn. (13).
4: **for** *iteration* = 1 to E **do**
5: Receive a code snippet C.
6: Sample an annotation $N \sim P(\cdot|C;\phi)$ according to Eqn. (8).
7: Receive the final reward $R(C, N)$.
8: Update the code annotation model (ϕ) using Eqn. (11).
9: Update the critic network (ρ) using Eqn. (13).
10: **end for**



$$r(s_t, n_t) = \begin{cases} \text{RetrievalReward}(C, n_{1..t}) & \text{if } n_t = \text{<EOS>} \\ 0 & \text{otherwise} \end{cases}$$

03 Testing Procedure of Coacor

Algorithm 2 : Generated Annotations for Code Retrieval.

Input: NL query Q, code snippet candidate C.

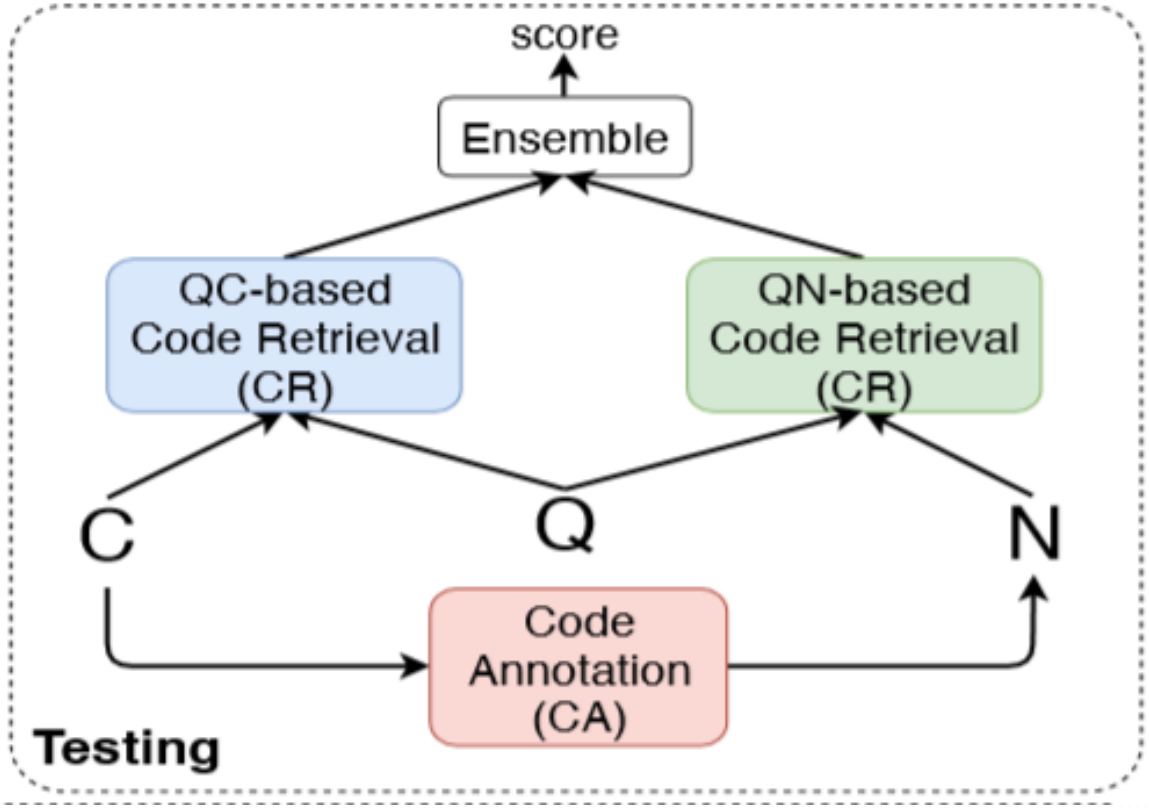
Output: The matching score, score(Q, C).

1: Receive score$_1(Q, C) = cos(v_q, v_c)$ from a QC-based code retrieval model.

2: Generate a code annotation $N \sim P(\cdot|C; \phi)$ via greedy search, according to Eqn. (8).

3: Receive score$_2(Q, C) = cos(v_q, v_n)$ from a QN-based code retrieval model.

4: Calculate score(Q, C) according to Eqn. (14).



$$\text{score}(Q, C) = \lambda * cos(v_q, v_n) + (1 - \lambda) * cos(v_q, v_c) \qquad (14)$$

04 Experiments

04 Dataset

StaQC: 119519 SQL<question title, code snippet> pairs mined from Stack Overflow

the largest-to-date in SQL domain; 75% training; 10% validation; 15% testing

Additional datasets for model comparison:

DEV and EVAL from Stack Overflow

Vocabulary Size:

7726 code tokens with average length of 60

7775 word tokens with average length of 9

The maximum length of the generated annotation in the experiments is set to 20

04 Evaluation Method

For each <NL query Q, code snippet C> pair in a dataset, take C as a positive code snippet.

Randomly sample K negative code snippets from all others except C in the dataset.

Calculate the rank of C among the K+1 candidates.

K is set to 49.

Mean Reciprocal Rank (MRR) metric:

$$\mathcal{D} = \{(Q_1, C_1), (Q_2, C_2), ..., (Q_{|\mathcal{D}|}, C_{|\mathcal{D}|})\}:$$

$$MRR = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \frac{1}{Rank_i}$$

The higher, the better !

04 Methods to Compare

Existing Baselines:

1.Deep Code Search (DCS): slightly modify the original model due to the lack of features like function names and API sequences.

2.Code-NN: one of the state-of-the-art models for both code annotation and code retrieval (LSTM + attention).

QN-based CR Variants:

1.QN-MLE: generate code annotations by maximizing the likelihood of a human-provided one.

2.QN-RL(BLUE): by maximizing the BLEU scores.

3.QN-RL(MRR): by maximizing the proposed MRR scores.

Ensemble CR Variants: *QN-MLE/BLUE/MRR + DCS/Code-NN*

04 Results

Research Questions:

- **RQ1 (CR improves CA)**: Is the proposed retrieval reward-driven CA model capable of generating rich code annotations that can be used for code retrieval (i.e., can represent the code snippet and distinguish it from others)?
- **RQ2 (CA improves CR)**: Can the generated annotations further improve existing QC-based code retrieval models?

Model	DEV	EVAL	StaQC-val	StaQC-test
Existing (QC-based) CR Baselines				
DCS [13]	0.566	0.555	0.534	0.529
CODE-NN [21]	0.530	0.514	0.526	0.522
QN-based CR Variants				
QN-CodeNN	0.369	0.360	0.336	0.333
QN-MLE	0.429	0.411	0.427	0.424
QN-RL^{BLEU}	0.426	0.402	0.386	0.381
QN-RL^{MRR} (ours)	0.534	0.512	0.516	0.523
Ensemble CR Variants				
QN-CodeNN + DCS	0.566	0.555	0.534	0.529
QN-MLE + DCS	0.571	0.561	0.543	0.537
QN-RL^{BLEU} + DCS	0.570	0.559	0.541	0.534
QN-RL^{MRR} + DCS (ours)	0.582*	**0.572***	0.558*	0.559*
QN-RL^{MRR} + CODE-NN (ours)	**0.586***	0.571*	**0.575***	**0.576***

RQ2: DCS improves 0.01~0.02 MRR ! Code-NN improves 0.03 MRR !

RQ1: Surpass 0.1~0.2 MRR ! Reflect the semantic meaning of each code more precisely !

Table 1: The main code retrieval results (MRR). * denotes significantly different from DCS [13] in one-tailed t-test ($p < 0.01$).

04 Results

Two examples of code snippets and their annotations their annotations generated by different code annotation models.

MRR: more concrete and precise !

Words semantically aligned between the generated and the human-provided annotations are highlighted.

MRR covers more conceptual keywords semantically aligned with the three NL queries (i.e., "average", "difference", "group").

Model	Annotation
Example 1 from EVAL set	
SQL Code	SELECT col3, Format(Avg([col2]-[col1]),"hh:mm:ss") AS TimeDiff FROM Table1 GROUP BY col3;
Human-provided	(1) find the average time in hours , mins and seconds between 2 values and show them in groups of another column (2) group rows of a table and find average difference between them as a formatted date (3) ms access average after subtracting
CODE-NN	how do i get the average of a column in sql?
MLE	how to get average of the average of a column in sql
RL^{BLEU}	how to average in sql query
RL^{MRR}	average avg calculating difference day in access select distinct column value sql group by month mysql format date function?
Example 2 from StaQC-test set	
SQL Code	SELECT Group_concat(DISTINCT(p.products_id)) AS comma_separated, COUNT(DISTINCT p.products_id) AS product_count FROM ...
Human-provided	how to count how many comma separated values in a group_concat
CODE-NN	how do i get the count of distinct rows?
MLE	mysql query to get count of distinct values in a column
RL^{BLEU}	how to count in mysql sql query
RL^{MRR}	group_concat count concatenate distinct comma group mysql concat column in one row rows select multiple columns of same id result

The baseline methods cover a very limited amount of conceptual keywords (e.g., without mentioning the concept "subtracting").

04 Results

Model	DEV	EVAL	StaQC-val	StaQC-test
CODE-NN [21]	17.43	16.73	8.89	8.96
MLE	18.99	19.87	10.52	10.55
RL^{BLEU}	21.12	18.52	12.72	12.78
RL^{MRR}	8.09	8.52	5.56	5.60

Table 3: The BLEU score of each code annotation model.

Compared with BLEU, a (task-oriented) semantic measuring reward, such as retrieval-based MRR score, can better stimulate the model to produce detailed and useful generations.

BLEU is an unappropriated evaluation metric for generation tasks.

Using the performance of a relevant model to guide the learning of the target model can be generalized to many other scenarios, e.g., conversation generation, machine translation, etc.

04 Discussion

1. Explore more about machine-machine collaboration mechanisms.

(Multiple models for either the same task or relevant tasks can be utilized to provide different views of effective rewards to improve the final performance.)

2. Collect paraphrases of queries to form QN pairs to directly use a QN-based CR model or an ensemble CR model for rewarding the CA model.

3. Extend the Coacor framework to other programming languages, such as Python and C# .

THANK YOU!