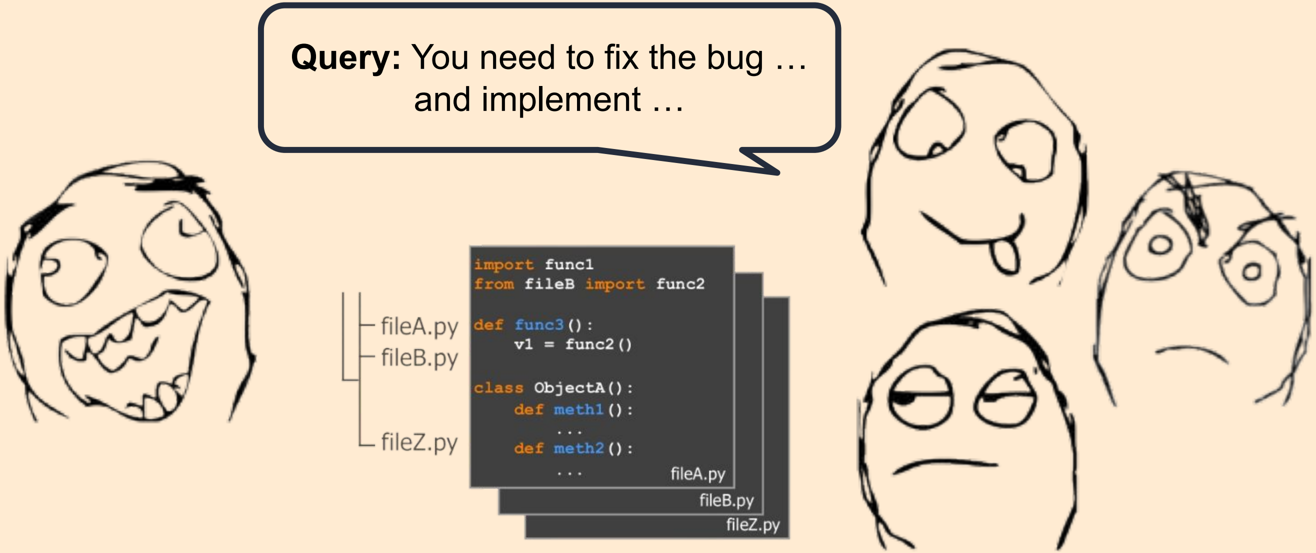
CoRet: Improved Retriever for Code Editing

Fabio J. Fehr, Prabhu Teja Sivaprasad, Luca Franceschi, Giovanni Zappella

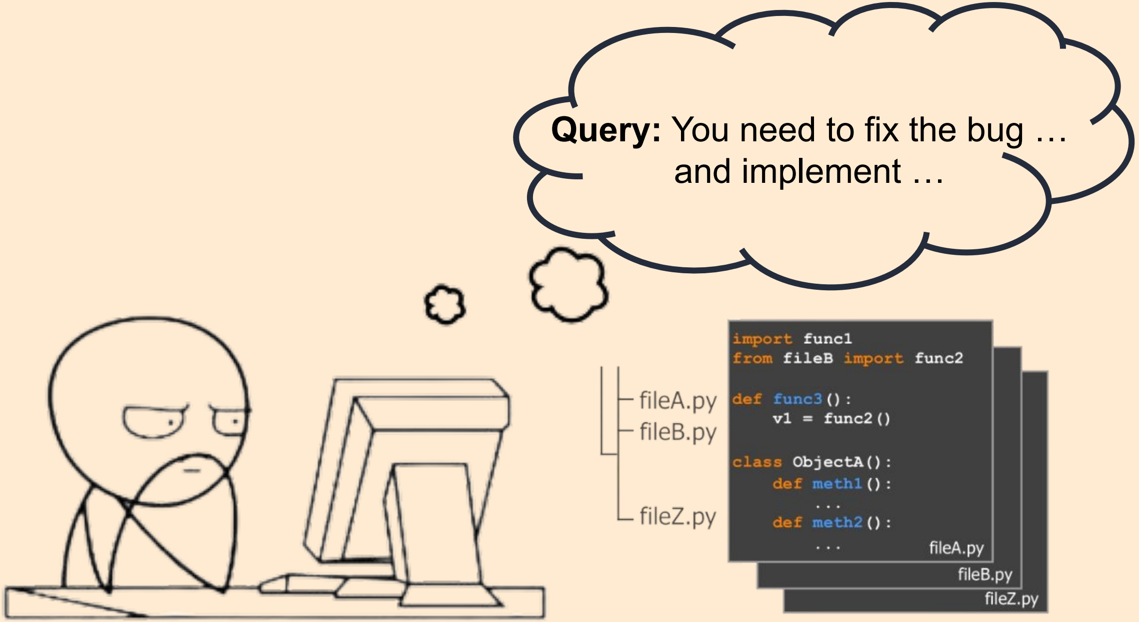
You

New Team


Query: You need to fix the bug …
and implement …
fileA.py
fileB.py
fileZ.py
import func1
from fileB import func2
def func3():
v1 = func2()
class ObjectA():
def meth1():
...
def meth2():
...
fileA.py
fileB.py
fileZ.py

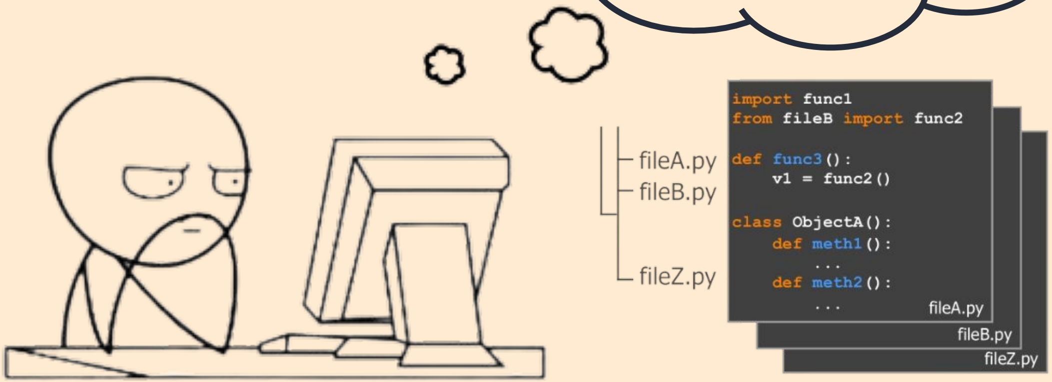
You

New Team


Query: You need to fix the bug …
and implement …
fileA.py
fileB.py
fileZ.py
import func1
from fileB import func2
def func3():
v1 = func2()
class ObjectA():
def meth1():
...
def meth2():
...
fileA.py
fileB.py
fileZ.py

You

Query: You need to fix the bug …
and implement …

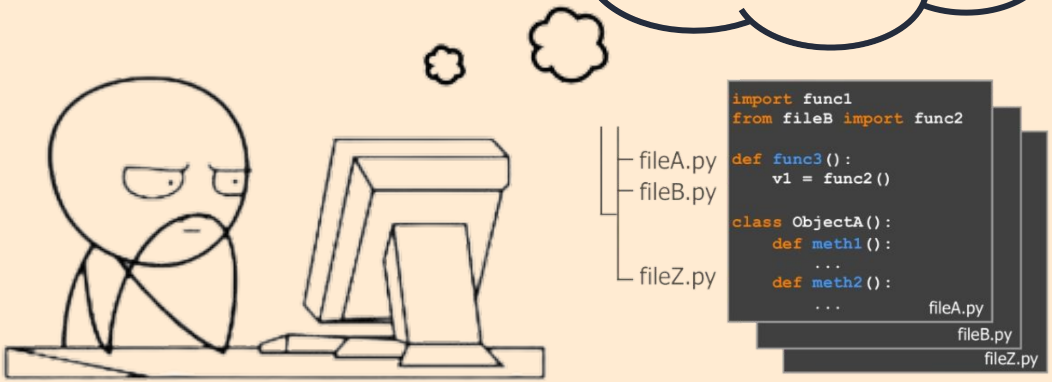


You

Which parts of the repo should you retrieve for editing?

Code Editing Retrieval Problem

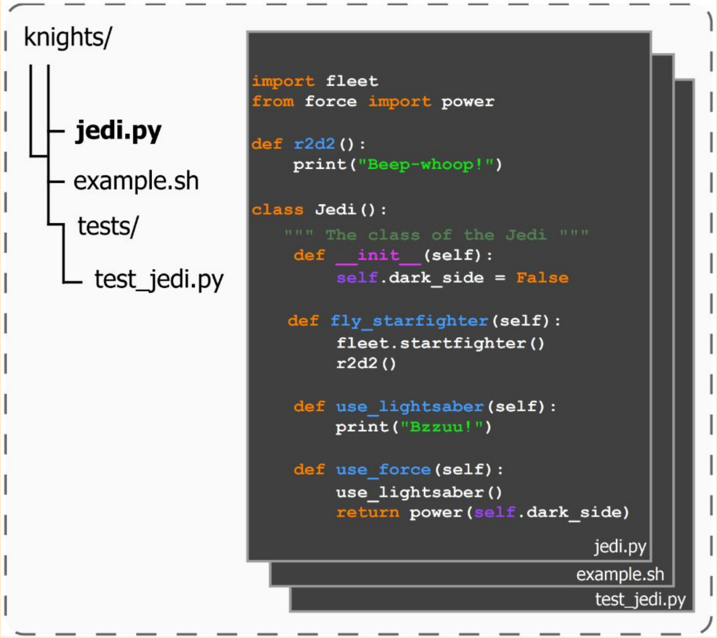
Query: You need to fix the bug …
and implement …



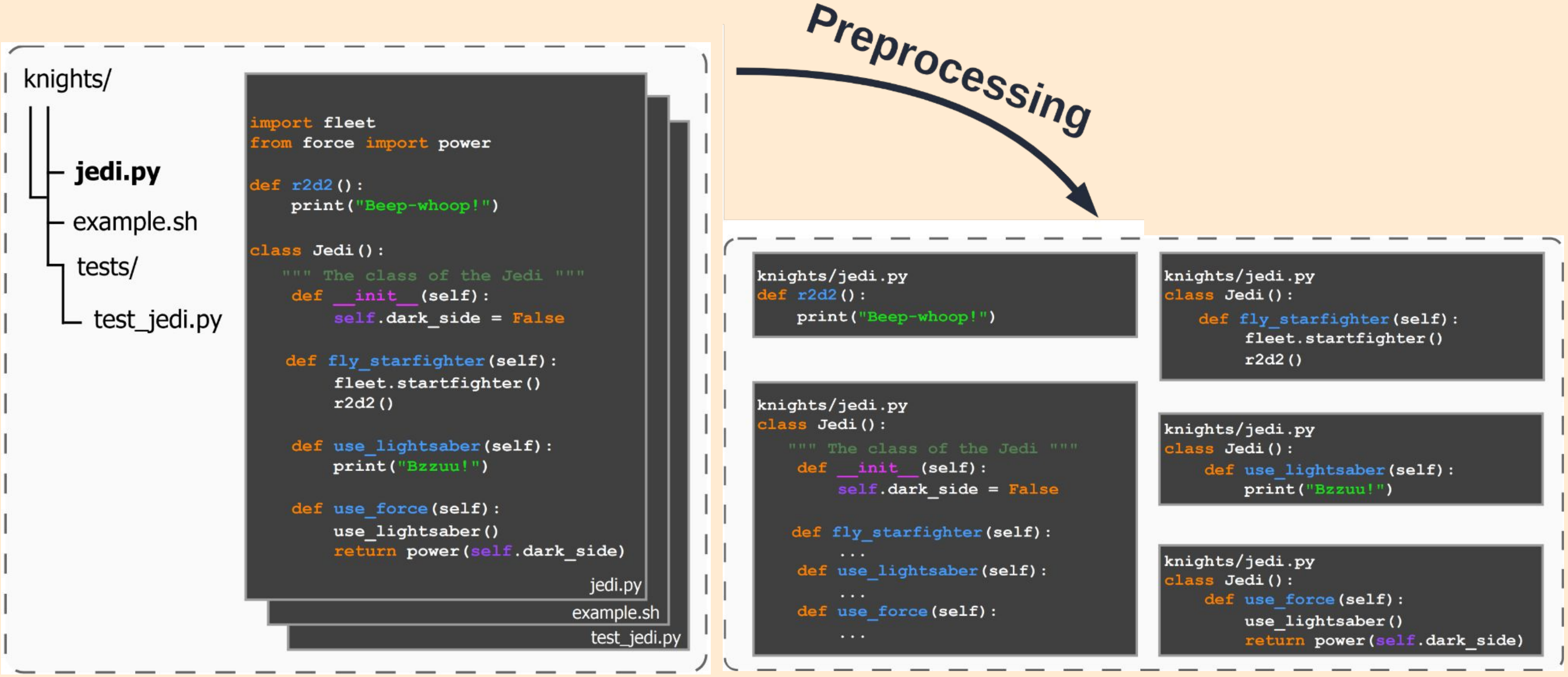
You

Which parts of the repo should you retrieve for editing?

Train a light-weight **code retriever** including **semantics** and **structure** across a repo.

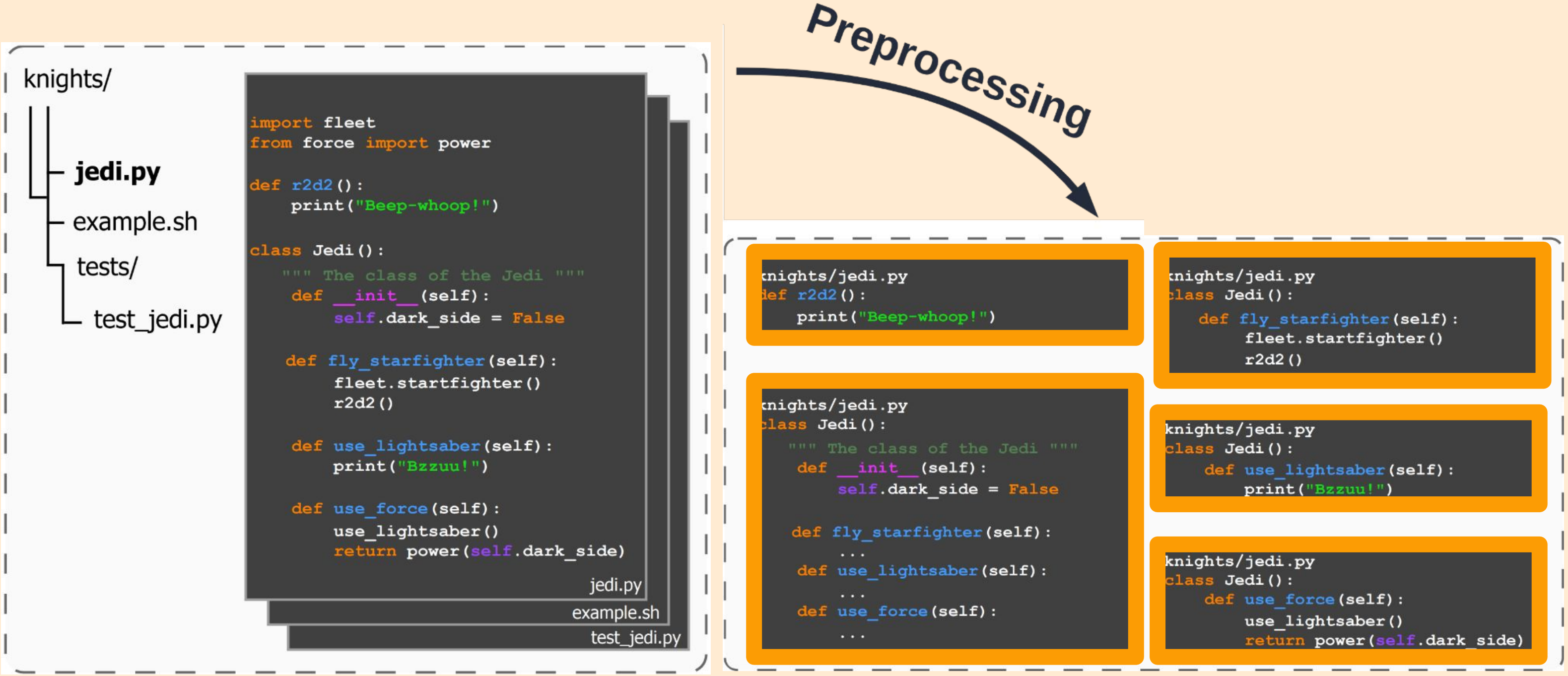


Code Repository



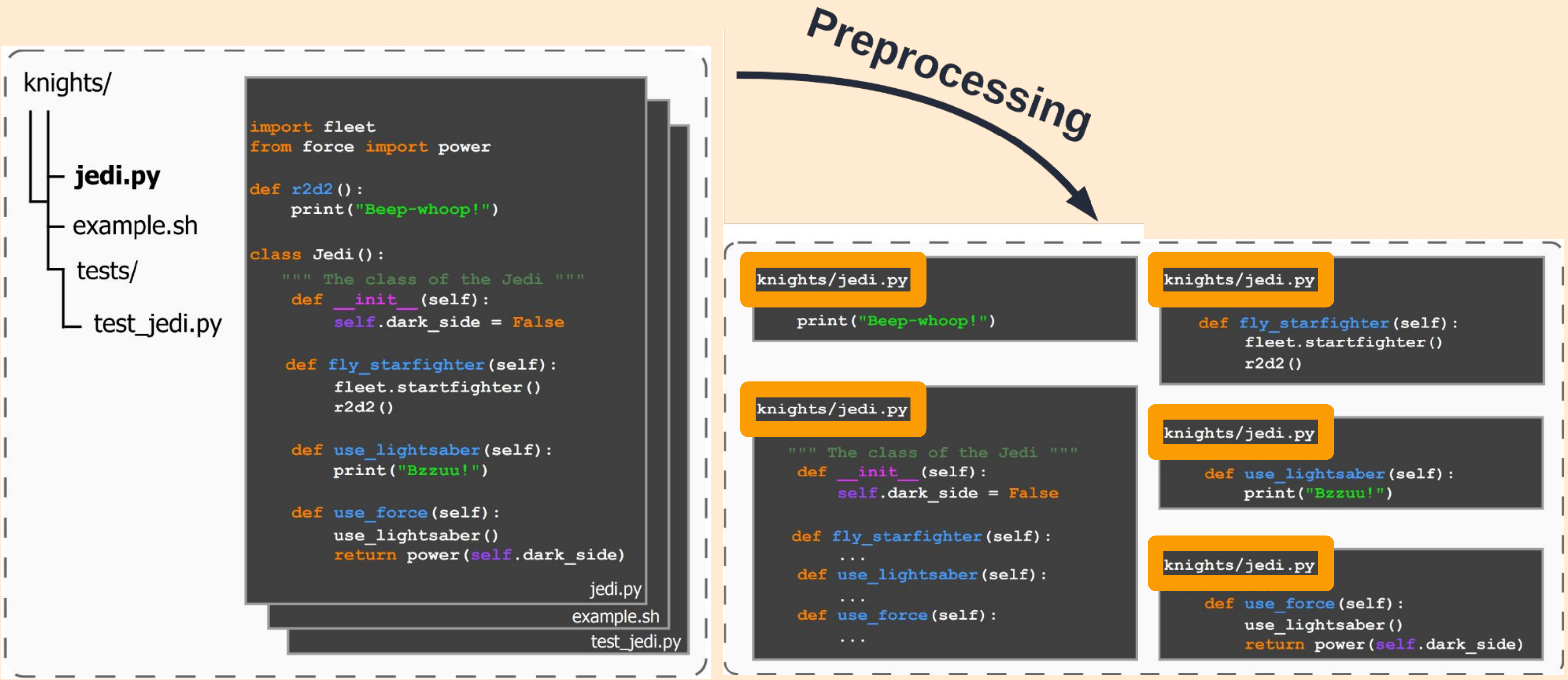
Code Repository **Code Chucks**

Code Chunks



Code Repository **Code Chucks**

Code Chunks with Repo-Hierarchy



Code Repository **Code Chucks**

```
knights/jedi.py
def lightsaber():
    lightsaber_on()
```

c_i

```
knights/utils.py
def lightsaber_on():
    print("Bzzuu!")
```

c_{out}

$$c_i \,;\, [\texttt{DOWN}] \,;\, c_{out}$$

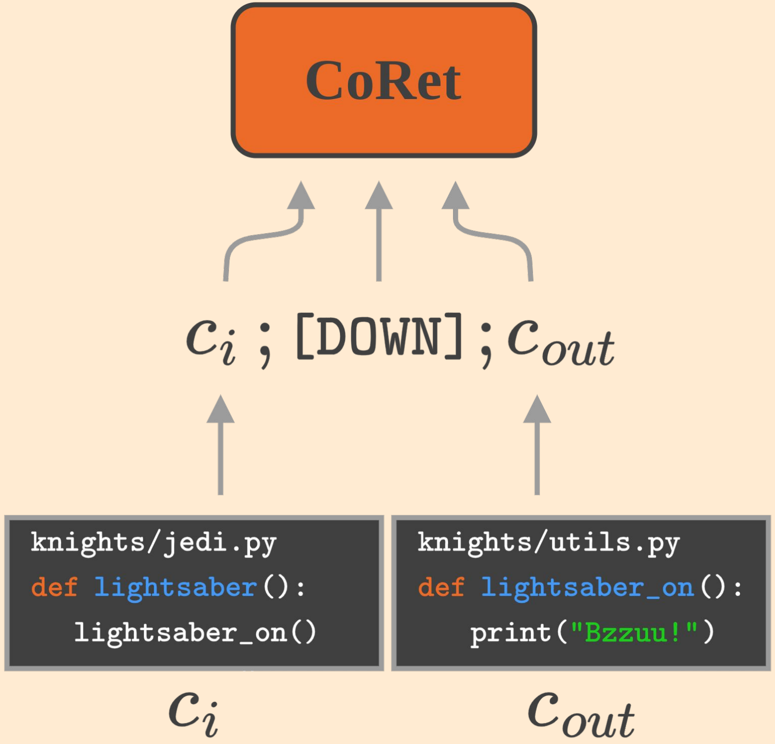
```
knights/jedi.py
def lightsaber():
    lightsaber_on()
```

c_i

```
knights/utils.py
def lightsaber_on():
    print("Bzzuu!")
```

c_{out}

Embedding with Call Graph Context



Training

$$\mathcal{L}(\theta)$$

Training

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i}^{N}$$

Training

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i}^{N} \frac{1}{\mathcal{C}_i^*} \sum_{c^* \in \mathcal{C}_i^*}$$

Training

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i}^{N} \frac{1}{\mathcal{C}_i^*} \sum_{c^* \in \mathcal{C}_i^*} \log \frac{\exp(\boldsymbol{q}_i \cdot \boldsymbol{c}^*)}{\exp(\boldsymbol{q}_i \cdot \boldsymbol{c}^*) + \sum_{c \in \mathcal{B}} \exp(\boldsymbol{q}_i \cdot \boldsymbol{c})}$$

Training with Likelihood Loss

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i}^{N}\frac{1}{\mathcal{C}_i^*}\sum_{c^*\in\mathcal{C}_i^*}\log\frac{\exp(\boldsymbol{q}_i\cdot\boldsymbol{c}^*)}{\exp(\boldsymbol{q}_i\cdot\boldsymbol{c}^*)+\boxed{\sum_{c\in\mathcal{B}}\exp(\boldsymbol{q}_i\cdot\boldsymbol{c})}}$$

Data & Model

SWE-Bench (Verified subset)

Data & Model

SWE-Bench (Verified subset)

LCA (Bug localisation task)

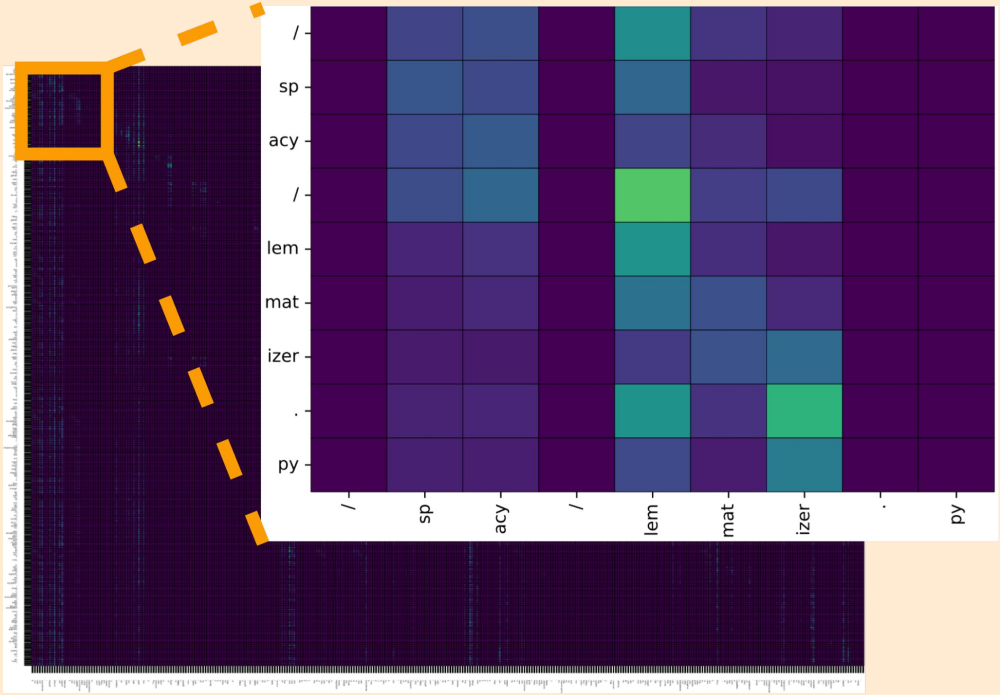
Data & Model

SWE-Bench (Verified subset)

LCA (Bug localisation task)

Codesage S (128M parameters)

Repo-Hierarchy is important



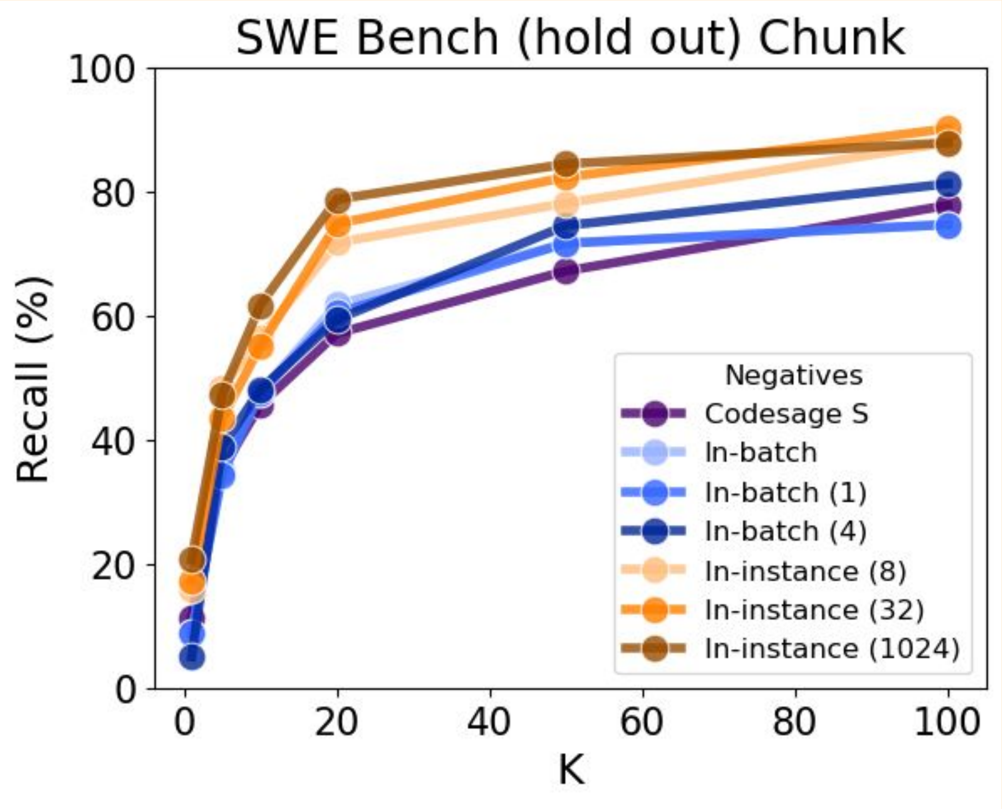
Call graph improves multi-chunk retrieval

	SWE Verified			LCA		
Model	@5	@20	MRR	@5	@20	MRR
CodeSage S	0.34	0.51	0.35	0.26	0.34	0.28
CoRet − CG	0.52	0.69	0.52	**0.32**	0.41	0.45
CoRet − CG + file	**0.54**	0.69	0.52	0.29	0.38	0.44
CoRet	**0.54**	**0.71**	**0.53**	**0.32**	**0.47**	**0.47**

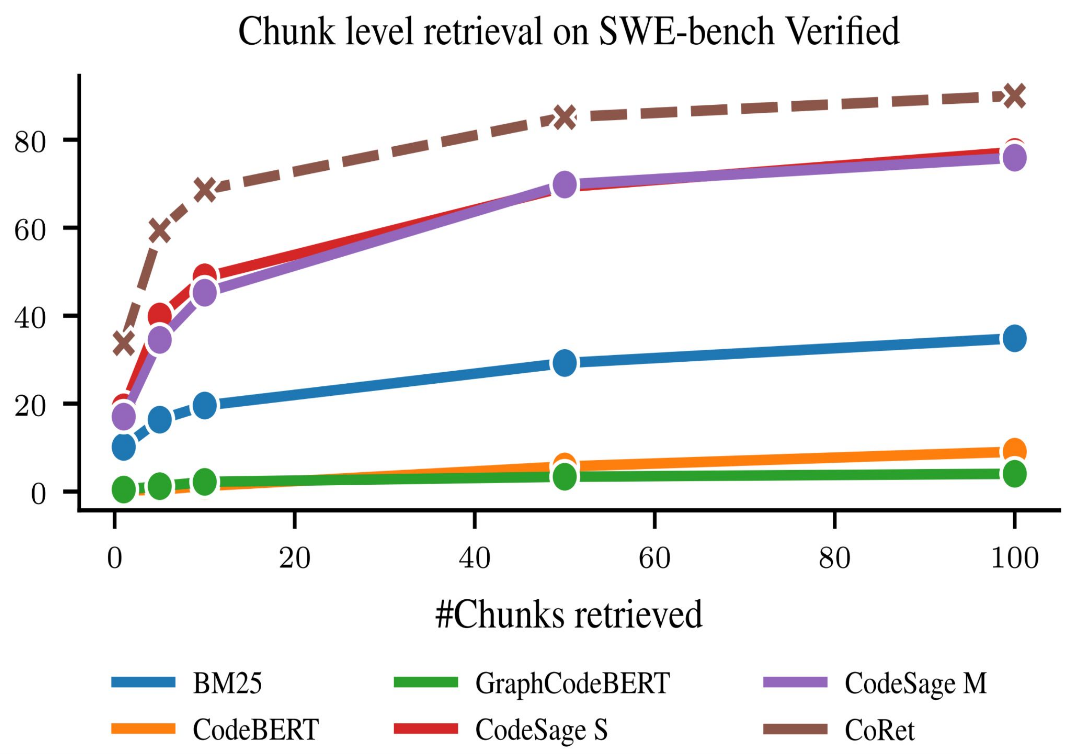
Call graph improves multi-chunk retrieval

	SWE Verified			LCA		
Model	@5	@20	MRR	@5	@20	MRR
CodeSage S	0.34	0.51	0.35	0.26	0.34	0.28
CoRet – CG	0.52	0.69	0.52	**0.32**	0.41	0.45
CoRet – CG + file	**0.54**	0.69	0.52	0.29	0.38	0.44
CoRet	**0.54**	**0.71**	**0.53**	**0.32**	**0.47**	**0.47**

Negatives from the same repo are best



Recall +15 percentage points!



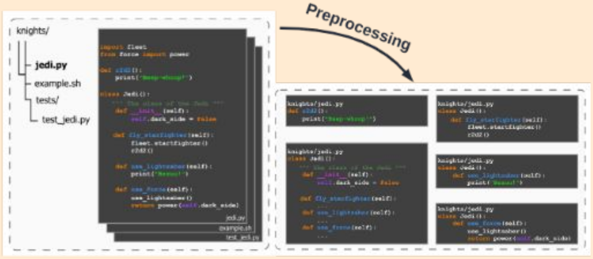
Train a light-weight **code retriever** including **semantics** and **structure** across a repo.

Paper

Code Editing Retrieval Problem

Which parts of the repo should you retrieve for editing?

Code Chunks with Repo-Hierarchy



The code repo is split into semantically succinct unit we called **code chunks**.
We include **repo-hierarchy** structure by including the file path string.

Embedding with Call Graph Context

c_i ; [DOWN] ; c_{out}

c_i:

```
knights/jedi.py
def lightsaber():
    lightsaber_on()
```

c_{out}:

```
knights/utils.py
def lightsaber_on():
    print("Bzzzu!")
```

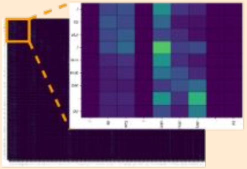
The chunk of interest c_i is concatenated as context with the out-going chunk c_{out} from the call graph and special token [DOWN].

Call graph improves multi-chunk retrieval

	SWE Verified			LCA		
Model	@5	@20	MRR	@5	@20	MRR
CodeSage S	0.34	0.51	0.35	0.26	0.34	0.28
CoRet − CG	0.52	0.69	0.52	**0.32**	0.41	0.45
CoRet − CG + file	**0.54**	0.69	0.52	0.29	0.38	0.44
CoRet	**0.54**	**0.71**	**0.53**	**0.32**	**0.47**	**0.47**

CoRet: Fine-tuned CodeSage S (130M parameters).
SWE Verified: Software Engineering Benchmark (Verified subset).
LCA: Long Code Arena (Bug localisation task).

Repo-hierarchy is important

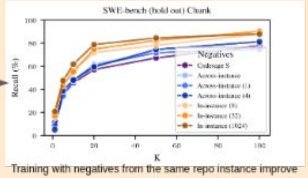


Training with Likelihood Loss

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i}^{N} \frac{1}{\mathcal{C}_i^*} \sum_{c^* \in \mathcal{C}_i^*} \log \frac{\exp(\boldsymbol{q}_i \cdot \boldsymbol{c}^*)}{\exp(\boldsymbol{q}_i \cdot \boldsymbol{c}^*) + \sum_{c \in \mathcal{B}} \exp(\boldsymbol{q}_i \cdot \boldsymbol{c})}$$

N = Number of repo instances i, $\mathcal{C}^*$ = Set of ground truth code chunks c^*, q = Natural Language query,
$\mathcal{B}$ = Random negative sample in the same repo instance.

Negatives from the same repo are best



Training with negatives from the same repo instance improve over negatives across repo instances (standard *in-batch* negatives).

Recall +15 percentage points