

# Good things come in three: Generating SO Post Titles with Pre-Trained Models, Self Improvement and Post Ranking

---

Duc Anh Le  
Hanoi University of Science and Technology  
Hanoi, Vietnam  
anh.ld204628@sis.hust.edu.vn

Phuong T. Nguyen  
Università degli studi dell'Aquila  
L'Aquila, Italy  
phuong.nguyen@univaq.it

Anh M. T. Bui  
Hanoi University of Science and Technology  
Hanoi, Vietnam  
anhbtm@soict.hust.edu.vn

Davide Di Ruscio  
Università degli studi dell'Aquila  
L'Aquila, Italy  
davide.diruscio@univaq.it



# Stack Overflow (SO)

- | 回答が得られる投稿はごく一部<sup>[48,50]</sup>
  - 投稿の情報が不足 or 他の開発者にとって魅力的でない<sup>[21]</sup>
- | 投稿の質の低さはタイトルの情報量の多さに起因<sup>[6,35]</sup>
  - 完結で意味のあるタイトルを記述する必要
    - 時間/労力がかかり，投稿全体を十分に理解する必要があり難しい

投稿内容からタイトルを自動生成する研究が盛ん

[48] Yu Zhou, Juanjuan Shen, Xiaoqing Zhang, Wenhua Yang, Tingting Han, and Taolue Chen. 2022. Automatic source code summarization with graph attention networks. *Journal of Systems and Software* 188 (2022), 111257.

[50] Wenhan Zhu, Haoxiang Zhang, Ahmed E. Hassan, and Michael W. Godfrey. 2022. An empirical study of question discussions on Stack Overflow. *Empir. Softw. Eng.* 27, 6 (2022), 148.

[21] Ke Liu, Guang Yang, Xiang Chen, and Chi Yu. 2022. SOTitle: A Transformer-based Post Title Generation Approach for Stack Overflow. In *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE Computer Society, 577–588.

[6] Denzil Correa and Ashish Sureka. 2013. Fit or unfit: analysis and prediction of closed questions' on stack overflow. In *Conference on Online Social Networks, COSN'13, Boston, MA, USA, October 7-8, 2013*, S. Muthu Muthukrishnan, Amr El Abbadi, and Balachander Krishnamurthy (Eds.). ACM, 201–212.

[35] László Tóth and László Vidács. 2019. Comparative Study of The Performance of Various Classifiers in Labeling Non-Functional Requirements. *Inf. Technol. Control.* 48, 3 (2019), 432–445.



# 先行研究 | SOの投稿タイトル自動生成

## | Code2Que<sup>[9]</sup>

- LSTMベースで、長いソースコード全体の意味を捉えられない可能性

## | M3NSCT5<sup>[46]</sup>

- CodeT5で、投稿に含まれるコードスニペットからタイトル生成
- コンテキストの欠如によって異なるタイトルを生成する可能性

## | SOTitle<sup>[21]</sup>

- T5で自然言語とコードスニペットからタイトル生成
- 各言語に対応したタイトルを生成可能

[21] Ke Liu, Guang Yang, Xiang Chen, and Chi Yu. 2022. SOTitle: A Transformer-based Post Title Generation Approach for Stack Overflow. In 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). IEEE Computer Society, 577–588.

[46] Fengji Zhang, Jin Liu, Yao Wan, Xiao Yu, Xiao Liu, and Jacky Keung. 2023. Diverse title generation for Stack Overflow posts with multiple-sampling-enhanced transformer. Journal of Systems and Software 200 (2023), 111672.



# 先行研究の課題

## 1. Exposure bias（曝露バイアス）

- 推論時の初期の予測の誤りが出力に大きな影響を与える可能性

## 2. モデルに内在するランダム性に依存

- 正解タイトルと大きく異なるタイトルが生成される場合がある

```
javascript
So I 'm trying to do a file upload using JQuery 's AJAX
thing , and it keeps giving me the error 500 . I am also
using this PHP code to process the file upload :
Unfortunately , I can not release the link to where the
actual problem is , but hopefully this code is enough to
help solve the problem . If any other details are needed ,
please do not hesitate to let me know
<code>
(function() {
  $('form').submit(function() {
    $.ajax({
      type: 'POST',
      url: 'photochallenge/submit.php',
      data: new FormData(this),
      processData: false,
      contentType: false,
      success: function(data) {
        Materialize.toast(data, 4000);
      }
    });
    return false;
  });
})();
```

Ground-Truth: JQuery AJAX File Upload Error 500

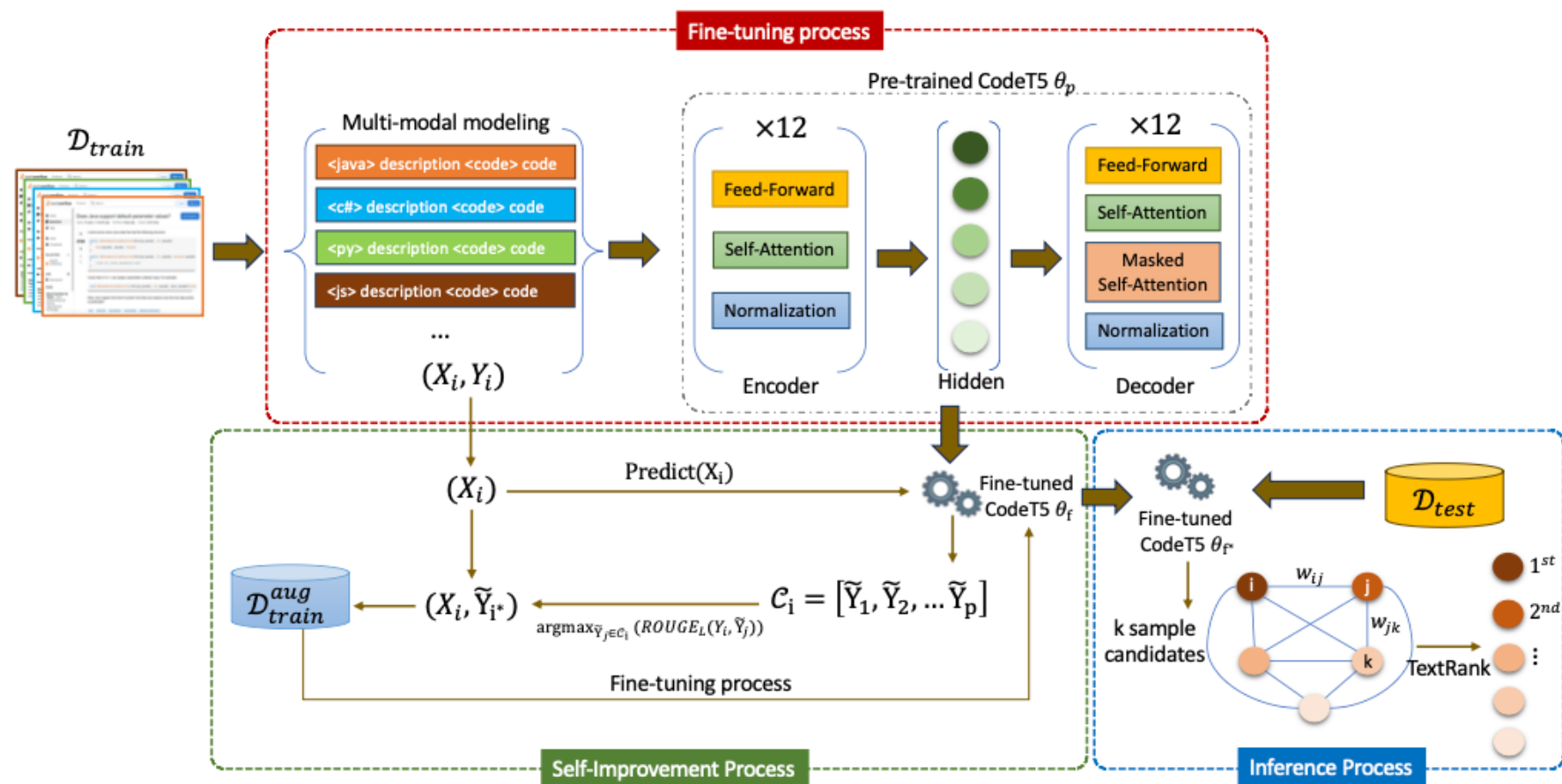
| # | Generated Title                           | Relevance Score |
|---|-------------------------------------------|-----------------|
| 1 | JQuery AJAX file upload gives 500         | 0.499           |
| 2 | JQuery AJAX file upload keeps getting 500 | 0.462           |
| 3 | JQuery AJAX file upload giving 500        | 0.499           |
| 4 | JQuery AJAX File Upload Error 500         | 0.999           |
| 5 | JQuery AJAX file upload returns 500       | 0.499           |
| 6 | JQuery AJAX file upload gives 500 error   | 0.462           |

2つの課題を克服する手法（FILLER）を提案



# FILLER

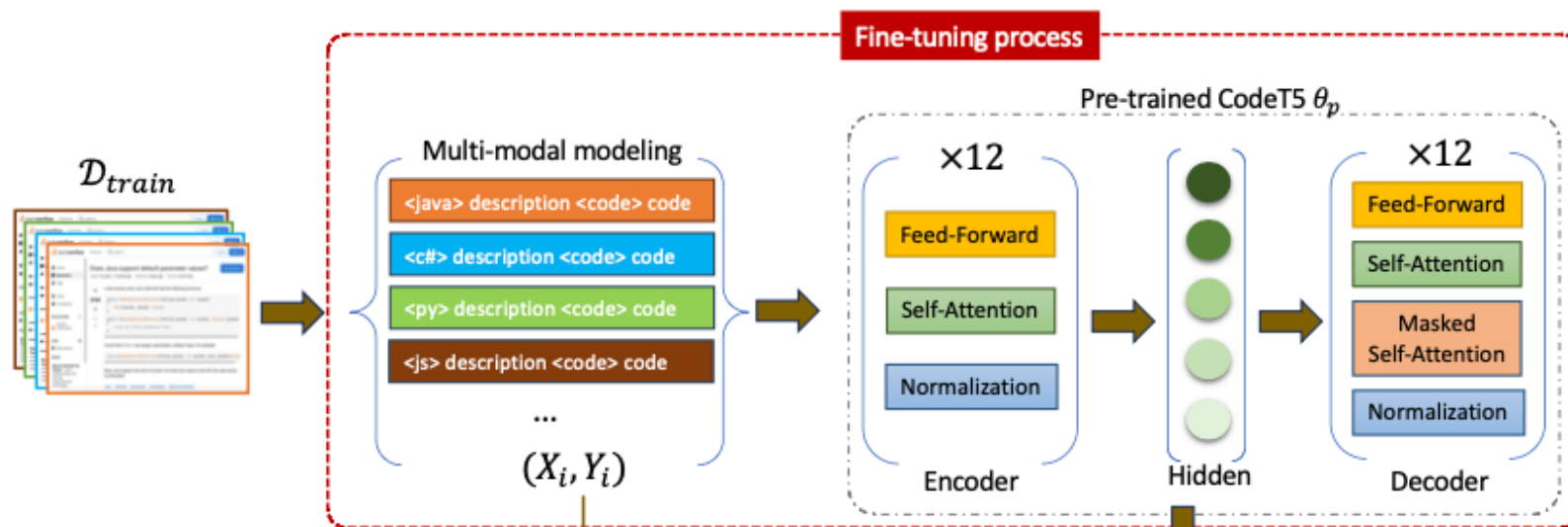
Fine-tuning, Self Improvement, 推論フェーズで構成





## Fine-tuning

- Input :  $\mathcal{X} = \langle \text{prefix} \rangle \oplus X_{desc} \oplus \langle \text{code} \rangle \oplus X_{code}$ 
  - $\langle \text{prefix} \rangle$ は各PLを示し,  $\langle \text{code} \rangle$ は区切り文字
- LOSS :  $\mathcal{L}_\theta = - \sum_{t=1}^{|Y|} \log P_\theta(y_t | y_{<t}; \mathcal{X})$ 
  - $\mathcal{L}_\theta$ は各PLの損失関数で, 正解タイトルトークン  $Y = \{t\}$ の負の最小尤度を最小化する関数

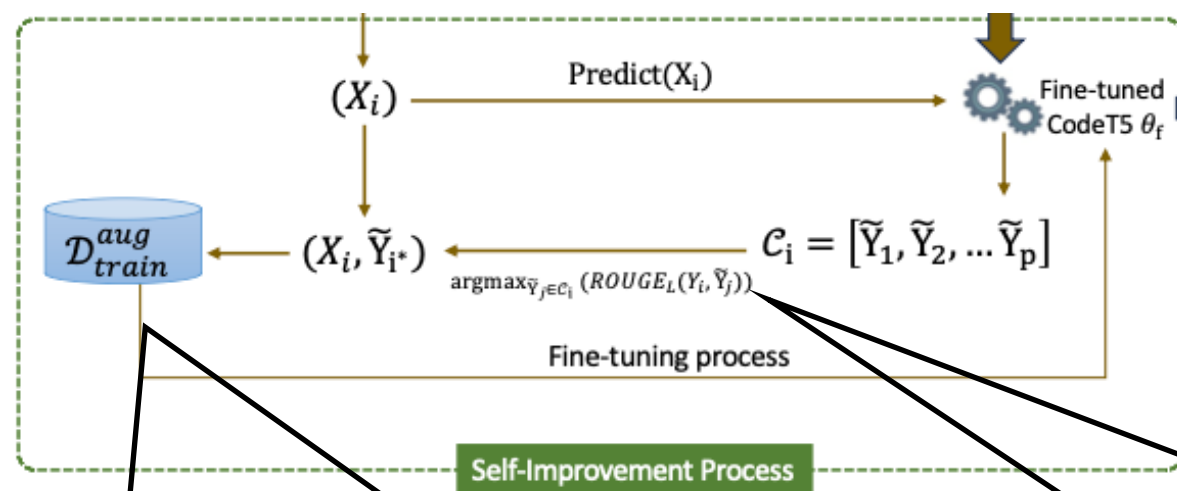




## Self Improvement

– Exposure biasを回避するための工夫

- オリジナルデータセットでFine-tuningしたCodeT5にSelf Improvementで拡張したデータセットで再度Fine-tuning



データオグメンテーションしたデータセットで再度Fine-Tuning

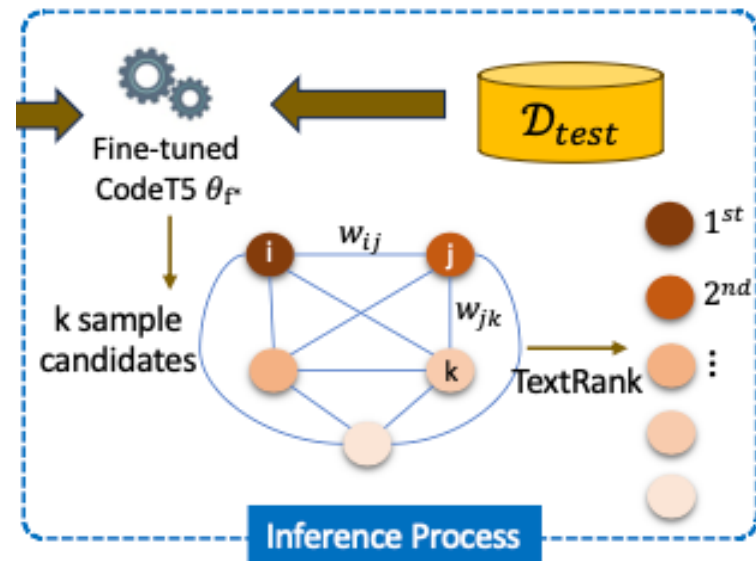
正解タイトルとのROUGE-Lスコアが最も高い候補を選択



## Post Ranking

### – ランダム性の回避

- 予測結果を複数出力し，最も品質の高いタイトルを選択



## TextRankアルゴリズムでランキング付け

### – 各ノードに関連性スコアを割り当て，スコアが収束するまで繰り返す

$$S_i = (1 - a) + a \sum_{j \neq i} \frac{\ell_{ij}}{\sum_{k \neq j} \ell_{jk}} S_j$$

- 各ノードは生成されたタイトル候補
- 各エッジの重みは2つのタイトル候補間のcos類似度（TF-IDF）



# Research Question

---

- | RQ1 : 事前学習済みモデルはSOの投稿タイトル生成の性能にどのような影響を与えるか
- | RQ2 : FILLERはSOの投稿タイトルを生成する上でSOTAと比較してどの程度効果的か
- | RQ3 : FILLERのどのフェーズが有効性に寄与するか
- | RQ4 : FILLERによって生成されたタイトルは他手法と比較して開発者にどのように認識されているか



# データセット

## | SOTitle<sup>[21]</sup>で使用されたベンチマークデータセット

– 4つの言語をそれぞれ訓練/評価/テストデータに分割

| Language     | Training | Validation | Testing |
|--------------|----------|------------|---------|
| Java         | 60,000   | 3,959      | 5,000   |
| C#           | 60,000   | 6,817      | 5,000   |
| Python       | 60,000   | 7,742      | 5,000   |
| JavaScript   | 60,000   | 5,780      | 5000    |
| <b>Total</b> | 240,000  | 24,298     | 20,000  |



# RQ1 | 事前学習済みモデルのタイトル生成性能

## Rougeスコアで評価

- $T5_{256+256}^{[21]}$  はコード256/自然言語256トークンを結合し入力
- $T5_{512}$  はコードと自然言語を結合して512トークンに削減
  - コードよりも自然言語に着目して削減

| Model                            | Java         |              |              | C#           |              |              | Python       |              |              | JavaScript   |              |              | Average      |              |              |
|----------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                                  | R-1          | R-2          | R-L          | R-1          | R-2          | R-L          | R-1          | R-2          | R-L          | R-1          | R-2          | R-L          | R-1          | R-2          | R-L          |
| <b><math>T5_{256+256}</math></b> | 26.49        | 9.80         | 24.60        | 26.94        | 10.76        | 25.17        | 29.00        | 10.79        | 26.70        | 28.53        | 10.77        | 26.47        | 27.74        | 10.53        | 25.74        |
| <b><math>T5_{512}</math></b>     | <b>28.72</b> | <b>11.14</b> | <b>26.59</b> | 29.03        | 12.07        | 27.21        | 31.39        | 12.37        | 28.87        | 30.96        | 12.65        | 28.80        | 30.03        | 12.06        | 27.87        |
| <b>BART</b>                      | 28.40        | 10.96        | 26.35        | 28.48        | 11.80        | 26.72        | 31.25        | 12.29        | 28.79        | 30.62        | 12.16        | 28.51        | 29.69        | 11.80        | 27.59        |
| <b>CodeBERT</b>                  | 25.22        | 9.31         | 23.62        | 25.17        | 10.28        | 23.85        | 28.16        | 10.75        | 26.36        | 27.49        | 10.67        | 25.85        | 26.51        | 10.25        | 24.92        |
| <b>UnixCoder</b>                 | 25.28        | 9.34         | 23.58        | 25.54        | 10.49        | 24.17        | 27.49        | 10.00        | 26.10        | 27.84        | 10.66        | 26.08        | 19.73        | 7.65         | 18.52        |
| <b>CodeT5</b>                    | 28.64        | 11.10        | 26.53        | <b>29.24</b> | <b>12.38</b> | <b>27.31</b> | <b>31.94</b> | <b>12.71</b> | <b>29.40</b> | <b>31.23</b> | <b>12.74</b> | <b>29.16</b> | <b>30.26</b> | <b>12.23</b> | <b>28.10</b> |

- CodeT5はより，コードスニペットの意味を理解できる

この結果から，FILLERをCode T5ベースで構築していく



# RQ2 | FILLERとSOTAの比較

Code2QueはLSTMベースでSeq2Seqはこのタスクに不向き

| Model        | Java  |       |       | C#    |       |       | Python |       |       | JavaScript |       |       | Average |       |       |
|--------------|-------|-------|-------|-------|-------|-------|--------|-------|-------|------------|-------|-------|---------|-------|-------|
|              | R-1   | R-2   | R-L   | R-1   | R-2   | R-L   | R-1    | R-2   | R-L   | R-1        | R-2   | R-L   | R-1     | R-2   | R-L   |
| Code2Que [9] | 25.25 | 9.17  | 23.57 | 26.55 | 10.57 | 25.01 | 27.93  | 10.36 | 25.87 | 27.50      | 10.40 | 25.70 | 26.81   | 10.13 | 25.04 |
| CCBERT [47]  | 26.25 | 9.65  | 24.57 | 26.00 | 10.76 | 24.72 | 28.55  | 10.69 | 26.54 | 28.34      | 10.83 | 26.64 | 27.29   | 10.48 | 25.62 |
| SOTitle [21] | 26.49 | 9.80  | 24.60 | 26.94 | 10.76 | 25.17 | 29.00  | 10.79 | 26.70 | 28.53      | 10.77 | 26.47 | 27.74   | 10.53 | 25.74 |
| M3NSCT5 [46] | 27.90 | 10.22 | 25.52 | 28.77 | 11.55 | 26.53 | 31.35  | 11.80 | 28.44 | 30.39      | 11.65 | 27.96 | 29.60   | 11.31 | 27.11 |
| GPT3.5-turbo | 25.68 | 7.62  | 22.52 | 25.21 | 7.80  | 22.69 | 28.48  | 8.92  | 25.06 | 27.15      | 8.19  | 24.11 | 26.63   | 8.13  | 23.60 |
| FILLER       | 30.48 | 11.52 | 27.94 | 30.89 | 12.81 | 28.60 | 33.64  | 13.00 | 30.63 | 33.14      | 13.24 | 30.55 | 32.04   | 12.64 | 29.43 |

T5ベース (SOTitle / M3NSCT5/ FILLER) が性能が高い

重要なキーワードは入っているが  
問題の細かなニュアンスが欠落

長いタイトルが生成されることが  
多々ある

元タイトルと近いタイトルを生成

```
I have java code that uses SortedMap.tailMap . In my ported code , I have  
SortedMap = Dictionary < IComparable , value > . I need a way to copy / mimic  
tailMap in C # . I 've thought something like the following : myDictionary.Where  
( p => p.Key.CompareTo ( value ) > = 0 ) .ToDictionaryThis returns a Dictionary  
, and I need a SortedDictionary returned . I could create a SortedDictionary from  
the Dictionary , but I feel like there should be a more elegant and performant  
way to do this as it 's already sorted.Another thought was to do something like  
That should work , I 'm not sure how the adding of values in a sorted order will  
affect timing on the inserts as I build that list.Any other thoughts ? Java  
SortedMapC # SortedDictionary
```

```
<code>  
var newSD = new SortedDictionary < k , v > ( ) ;  
foreach ( var p in oldDictionary.Where ( p => p.Key.CompareTo ( value ) > = 0 ) )  
newSD.Add ( p.Key , p.Value ) ;
```

| Ground-Truth | Equivalent of Java 's SortedMap.tailMap in C # SortedDictionary                            |
|--------------|--------------------------------------------------------------------------------------------|
| Code2Que     | How to copy / mimic tailMap in C # ?                                                       |
| CCBERT       | Copy / mimic tailMap in C #                                                                |
| SOTitle      | How to copy / mimic tailMap in C # ?                                                       |
| M3NSCT5      | How to copy a Dictionary < IComparable , value > to<br>SortedDictionary < k , v > in C # ? |
| FILLER       | C # equivalent of Java 's SortedMap.tailMap                                                |



# RQ3 | FILLERの各フェーズの有効性

## | アブレーションスタディを実施

- Self ImprovementとPost Rankingが両方ない場合は性能が低下
  - いずれかを除外するだけでも性能に悪影響が出る

| Model                       | Java  |       |       | C#    |       |       | Python |       |       | JavaScript |       |       | Average |       |       |
|-----------------------------|-------|-------|-------|-------|-------|-------|--------|-------|-------|------------|-------|-------|---------|-------|-------|
|                             | R-1   | R-2   | R-L   | R-1   | R-2   | R-L   | R-1    | R-2   | R-L   | R-1        | R-2   | R-L   | R-1     | R-2   | R-L   |
| FILLER <sub>w/o SI+PR</sub> | 28.64 | 11.10 | 26.53 | 29.24 | 12.38 | 27.31 | 31.94  | 12.71 | 29.40 | 31.23      | 12.74 | 29.16 | 30.26   | 12.23 | 28.10 |
| FILLER <sub>w/o PR</sub>    | 28.93 | 11.26 | 26.89 | 29.27 | 12.46 | 27.45 | 32.17  | 12.78 | 29.64 | 31.77      | 12.89 | 29.59 | 30.54   | 12.35 | 28.39 |
| FILLER <sub>w/o SI</sub>    | 29.71 | 11.29 | 27.35 | 29.86 | 12.46 | 27.78 | 33.05  | 12.97 | 30.20 | 32.34      | 12.90 | 29.95 | 31.24   | 12.41 | 28.82 |
| FILLER                      | 30.48 | 11.52 | 27.94 | 30.89 | 12.81 | 28.60 | 33.64  | 13.00 | 30.63 | 33.14      | 13.24 | 30.55 | 32.04   | 12.64 | 29.43 |

- Post RankingにおけるTop-kの有効な閾値

| Top-k Ranking | Average |       |       |
|---------------|---------|-------|-------|
|               | R-1     | R-2   | R-L   |
| K = 10        | 31.46   | 12.45 | 29.02 |
| K = 20        | 31.82   | 12.55 | 29.23 |
| K = 30        | 32.04   | 12.64 | 29.43 |
| K = 40        | 32.06   | 12.62 | 29.44 |
| K = 50        | 32.13   | 12.56 | 29.45 |



# RQ4 | 開発者による評価

## 200個のタイトルをランダムサンプリング

– 修士学生の学生3名が可読性(R1)と関連性(R2)を1～4で評価

| No. | Ground-truth title                                                                  | SOTitle                                                       |                |                | GPT3.5-turbo                                                                                                                       |                |                | FILLER                                                                                      |                |                |
|-----|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------|----------------|------------------------------------------------------------------------------------------------------------------------------------|----------------|----------------|---------------------------------------------------------------------------------------------|----------------|----------------|
|     |                                                                                     | Generated title                                               | R <sub>1</sub> | R <sub>2</sub> | Generated title                                                                                                                    | R <sub>1</sub> | R <sub>2</sub> | Generated title                                                                             | R <sub>1</sub> | R <sub>2</sub> |
| 1   | React-highcharts scrollbar feature not working                                      | react-highchart scrollbar not working                         | 2              | 4              | How to handle errors and rewrite mapping logic using Ramda in functional programming for indexing target columns in CSV header?    | 4              | 1              | react-highcharts bar scrollbar not working                                                  | 2              | 4              |
| 2   | Why is 'this' undefined in the debugger but printable in a console.log?             | Why does console.log set a breakpoint on console.log?         | 3              | 1              | What is the purpose of the "(0, foo)()" syntax in the JavaScript code snippet provided?                                            | 3              | 1              | Why is 'this' undefined in console.log?                                                     | 4              | 3              |
| 3   | Why does an unhandled exception in this background thread not terminate my process? | Unhandled exception in foreground and background threads      | 4              | 2              | How to achieve constant rotation of a character's head without flipping the character when rotated more than 180 degrees in Unity? | 3              | 1              | Why does an unhandled exception in a background thread not cause the process to terminate ? | 3              | 4              |
| 4   | "list.toArray (T [ ] a), what if the "T" is a "runtime type"?"                      | How to convert a list into an array if it is a runtime type?" | 3              | 2              | Java 9 issue with casting toArray in Arrays.asList causing code to not work                                                        | 3              | 1              | List.toArray (T [ ] a), what if the T is a "runtime type"?"                                 | 2              | 4              |

– それぞれのツール間で人間評価に統計的有意差があるか検定

- WilCoxon( $p \leq 0.05$ )で有意差有り



# まとめ & 所感

## | まとめ

- 魅力的なSO投稿タイトル生成
- Self ImprovementとPost Rankingによる性能向上

## | 所感

- 妥当なアプローチであり，複数言語に対応できるようになった点も貢献として大きい
- TextRankを利用した理由を知りたい．他のアルゴリズムと比べて優位な点を知りたかった