

Um estudo preliminar sobre o uso de uma arquitetura deep learning para seleção de respostas no problema de recuperação de código-fonte

VII Workshop on Software Visualization, Evolution and Maintenance

Marcelo de Rezende Martins¹, Marco Aurélio Gerosa²

¹Instituto de Pesquisas Tecnológicas (IPT)

²Northern Arizona University (NAU)

1. Introdução

2. Abordagem

3. Experimento

Avaliação

Introdução

A tarefa do *code retrieval* ou recuperação de trecho de código-fonte consiste em:

Dado uma descrição em linguagem natural, recuperar o trecho de código-fonte mais relevante, tal que os desenvolvedores possam encontrar rapidamente os trechos de código que atendam as suas necessidades. [2]

Code Retrieval: Dada uma questão em linguagem natural Q , um modelo F_r irá aprender a recuperar o trecho de código-fonte $C^* \in \mathbb{C}$ com a maior pontuação [6]:

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

Abordagem

Hipótese inicial: *software é uma forma de comunicação humana e tem propriedades estatísticas similares a corpora de linguagem natural [1]*

Seja \mathbb{Q} o conjunto formado pelas questões e \mathbb{C} o conjunto composto por trechos de código-fonte [3]:

$$\mathbb{Q} \xrightarrow{f} \mathbb{V}_q \rightarrow h_{\theta}(\mathbb{V}_q, \mathbb{V}_c) \leftarrow \mathbb{V}_c \xleftarrow{g} \mathbb{C} \quad (2)$$

Função de perda *hinge* [5, 3]:

$$L = \max(0, m - h_{\theta}(q_i, c_i^+) + h_{\theta}(q_i, c_i^-)) \quad (3)$$

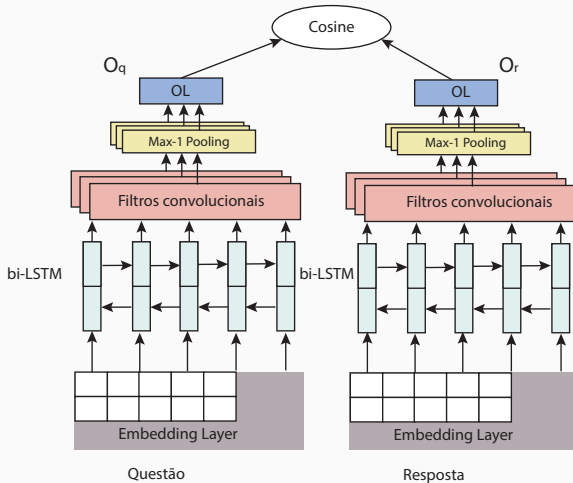


Figure 1: Figura adaptada do [5].

Experimento

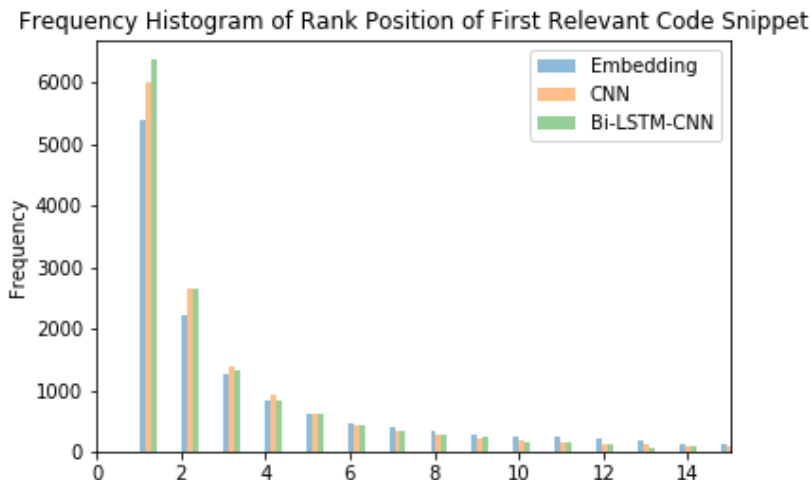
Amostras	Quantidade de (q_i, c_i^+)
Treinamento	60.083
DEV	1.085
EVAL	1.084
Total	62.252

Table 1: Divisão das amostras para treinamento conforme os critérios adotados por [4].

Modelos	Resultados (MRR)
Embedding	$0,52 \pm 0,01$
CNN	$0,58 \pm 0,01$
bi-LSTM-CNN	$0,60 \pm 0,02$

Table 2: Resultado preliminar do modelo bi-LSTM-CNN proposto em comparação a outros dois modelos (CNN e Embedding). Estes resultados foram obtidos a partir da amostra EVAL.

Histograma das posições dos trechos de código-fonte relevantes



Python and appending items to text and excel file¹

BiLSTM-CNN

```
Yvalues = [1, 2, 3, 4, 5]
file_out = open('file.csv', 'wb')
mywriter = csv.writer(file_out, delimiter = '\n')
mywriter.writerow(Yvalues)
file_out.close()
```

CNN

```
import csv

with open("output.csv", "wb") as f:
    writer = csv.writer(f)
    writer.writerow(a)
```

¹<https://stackoverflow.com/questions/24593478/python-and-appending-items-to-text-and-excel-file>

Perguntas?



M. Allamanis, E. T. Barr, P. Devanbu, and C. Sutton.

A survey of machine learning for big code and naturalness.

ACM Comput. Surv., 51(4):81:1–81:37, July 2018.



Q. Chen and M. Zhou.

A neural framework for retrieval and summarization of source code.

In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, ASE 2018, pages 826–831, New York, NY, USA, 2018. ACM.



X. Gu, H. Zhang, and S. Kim.

Deep code search.

In *Proceedings of the 40th International Conference on Software Engineering, ICSE '18*, pages 933–944, New York, NY, USA, 2018. ACM.



S. Iyer, I. Konstas, A. Cheung, and L. Zettlemoyer.

Summarizing source code using a neural attention model.

In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2073–2083, Berlin, Germany, Aug. 2016. Association for Computational Linguistics.



M. Tan, C. dos Santos, B. Xiang, and B. Zhou.

Lstm-based deep learning models for non-factoid answer selection.

CoRR, abs/1511.04108, 2015.



Z. Yao, D. S. Weld, W.-P. Chen, and H. Sun.

Staqc: A systematically mined question-code dataset from stack overflow.

In *Proceedings of the 2018 World Wide Web Conference, WWW '18*, pages 1693–1703, Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee.