StackReQA: Stack Overflow Retrieval Question Answering

Alyssa Augsburger
John Lee
Sanjay Saravanan

April 15, 2021

Introduction

Goal

- Identify the best Stack Overflow answer to a proposed question

Approach

- Represent text and code with different embeddings
- Apply different ordering techniques
- Utilize different architectures

Question

How to display 5 numbers per line from a list?
lx = [1,2,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25]
def display(lx):
for i in range(0,len(lx), 5):
x = lx[i:i +5]
return x
print(display(lx))
my current code is displaying only one line that contains 5 numbers, and the expected should be 5 lines that contains 5 numbers per line

Answer

You can make the function a generator by making it yield the sliced lists instead:
def display(lx):
for i in range(0, len(lx), 5):
yield lx[i:i + 5]
print(*display(lx), sep='\\n')
This outputs: [1, 2, 4, 5, 6] [7, 8, 9, 10, 11] [12, 13, 14, 15, 16] [17, 18, 19, 20, 21] [22, 23, 24, 25] "]

Background

- ReQA: An Evaluation for End-to-End Answer Retrieval Models
  Ahmad et. al., 2019
  - End-to-end solution
  - Model should be contextually aware

- Semantic Similarity and Response Prediction for Programming QA
  Kulkarni and Rosich, 2020
  - Solved a similar problem, but excluded code snippets

- MultiReQA: A Cross-Domain Evaluation for Retrieval Question Answering Models
  Guo et. al., 2020
  - Evaluated retrieval tasks using supervised neural models based on BERT and USE-QA

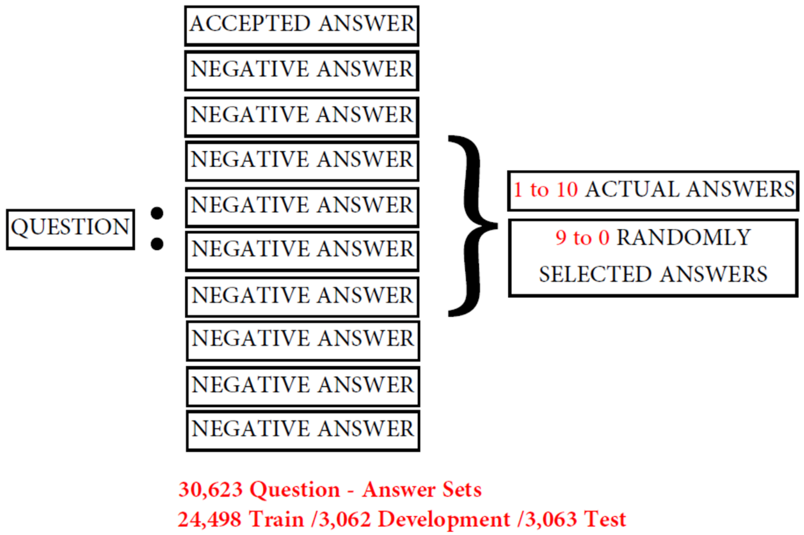
Task & Data

Task

Given 1 question, pick best of 10 answers

Data

- 30,623 Unique Questions
- 60,335 Answers
- <5% without code
- <5% without text



Evaluation & Baseline

Evaluation

- Mean Reciprocal Rank (MRR)
- Precision

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}$$

Baseline

- BM-25

Miscellaneous Experiments [1]	MRR	Precision
BM-25 Baseline		
BM-25 With Text Only	0.4141	0.2151
BM-25 With Text And Code	**0.6392**	**0.4757**

Strategy - Representation

- Universal Sentence Encoder (USE)
- Bidirectional Encoder Representation for Transformers (BERT)
- Word2Vec

Embeddings Experiment	MRR	Precision
USE - Data As Is	0.7567	0.6108
USE - Averaged Sentences	0.6535	0.4900
BERT - Last 512 Tokens	0.5656	0.5856
BERT - Last 136/87 Tokens	0.5633	0.3939
BERT - First and Last 136/87 Tokens	0.5862	0.4140
Word2Vec - Code Reserved Words	0.2317	0.0200
Word2Vec - Top 50 Code	0.3214	0.0797
Word2Vec - Text and Code	0.5170	0.3475

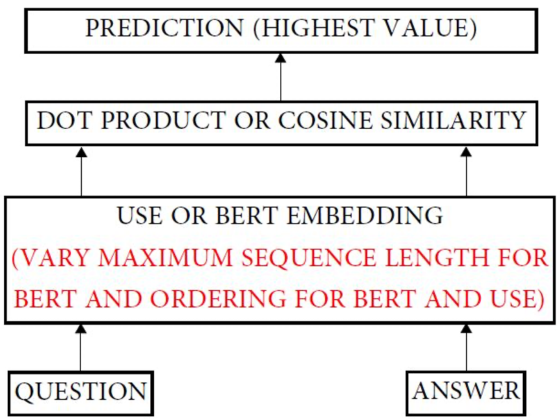
Strategy - Ordering

Embeddings Experiment [1]	MRR	Precision
USE - Data As Is	**0.7567**	**0.6108**
USE - Averaged Sentences	0.6535	0.4900
USE - Text Then Code	0.7374	0.5856
USE - Code Then Text	0.7361	0.5814
BERT - Last 512 Tokens	0.5656	0.3939
BERT - Last 136/87 Tokens	0.5633	0.3963
BERT - First And Last 136/87 Tokens	0.5862	0.4140
BERT - First And Last 136/87 Tokens - Text Then Code	0.6165	0.4508
BERT - First And Last 136/87 Tokens - Code Then Text	0.6147	0.4461
BERT - First And Last 136/87 Tokens - Text Then Code - Hidden States	0.5975	0.4369
Word2Vec - Code Reserved Words	0.2317	0.0200
Word2Vec - Top 50 Code	0.3214	0.0797
Word2Vec - Text And Code	0.5170	0.3475

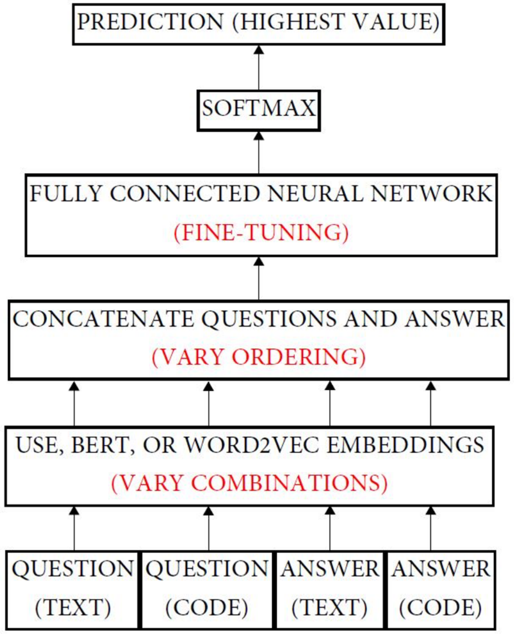
[1]Dot product is applied on USE embeddings. Cosine Similarity is applied on BERT and Word2Vec embeddings.

Ordering and Architecture Experiment	MRR	Precision
Concatenated Neural Network with Ordering Variations		
USE - Question + Answer	0.7794	0.6252
USE - Answer + Question	0.7323	0.5657
BERT - Question + Answer	0.6741	0.4904
BERT - Answer + Question	0.6814	0.5011
Two Towers		
USE	**0.7817**	**0.6290**
BERT	0.6960	0.5185

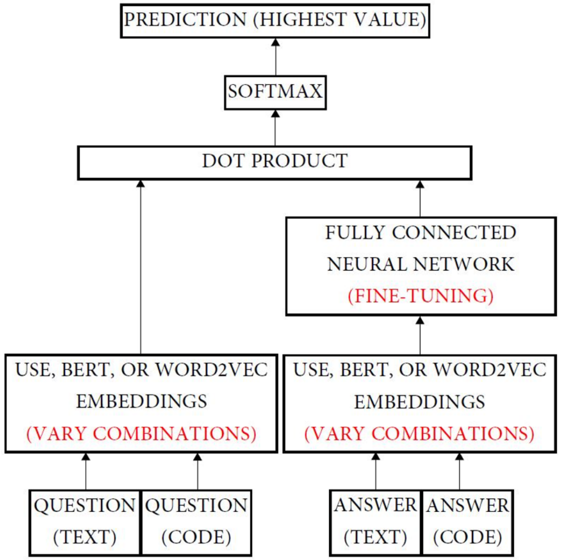
Strategy - Architectures



Dual Encoder



Concatenated Neural Network
(Early Fusion)

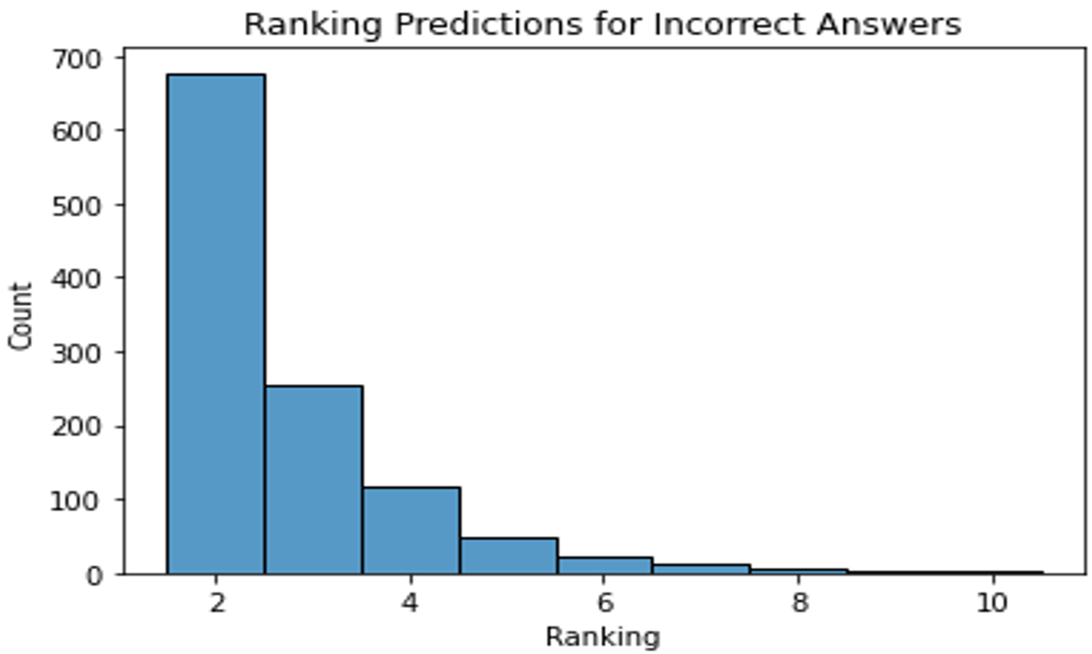


Two Towers
(Late Fusion)

Results & Error Analysis

Architecture	Best Models	MRR	Precision
Baseline	BM-25 - Text and Code	0.6392	0.4757
Dual Encoder	USE - Data As Is	0.7567	0.6108
Concatenated Neural Network	USE - Question Then Answer	0.7794	0.6252
Two Towers	USE	**0.7817**	**0.6290**

Table 5: Best performing models of each architecture.



Hyperparameters

- Single dense layer
- 512 nodes
- Adam Optimizer
- Learning Rate - 0.001
- Sparse categorical cross-entropy loss
- Batch size - 64
- 2 epochs

Conclusion

Experiments

- Representation (USE, BERT, Word2Vec)
- Ordering (Code and Text, Question + Answer, etc...)
- Architectures (BM-25, Dual Encoder, Early/Late Fusion)

Future Work

- Better representation of code data
- More fine-tuning
- Expanded dataset
- Ablation Study

Thank You.

References

- Amin Ahmad, Noah Constant, Yinfei Yang, and DanielCer. 2019. Reqa: An evaluation for end-to-end answer retrieval models.Proceedings of the 2nd Workshop on Machine Reading for Question Answering.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. CoRR, abs/1810.04805.
- Mandy Guo, Yinfei Yang, Daniel Cer, Qinlan Shen,and Noah Constant. 2020. Multireqa: A cross-domain evaluation for retrieval question answering models.
- Rahul Kulkarni and Ryan Rosich. 2020. Semantic similarity and response prediction for programming qa.
- Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.
- Rudi Seitz. 2020. Understanding tf-idf and bm25.
- Kaggle Stack Overflow. 2019. Stack overflow data.
- Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang.2019. How to fine-tune BERT for text classification?CoRR, abs/1905.05583.
- Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung,Brian Strope, and Ray Kurzweil. 2019. Multilingual universal sentence encoder for semantic retrieval