Question Answering using Deep Learning

Bo Liu Junzhang Li Guanghui Min Runwei Wang

Department of Statistics | EECS595 | Fall 2019

UNIVERSITY OF MICHIGAN

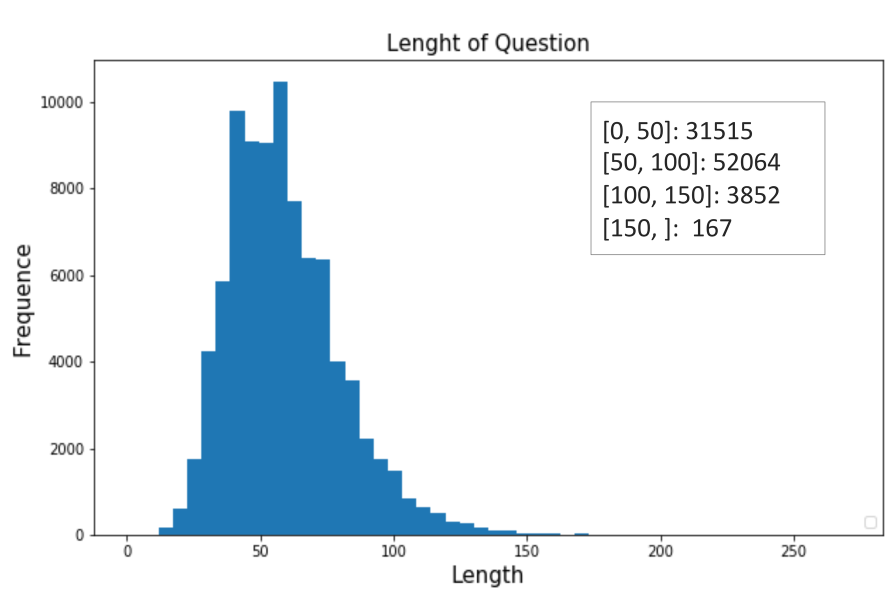
Background + Task

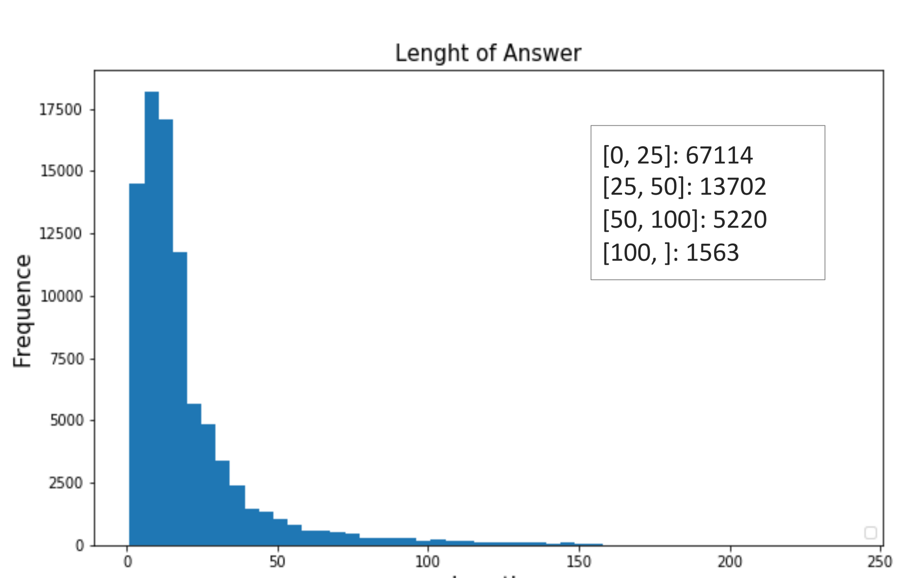
Background: Text understanding and reasoning have been popular topics in natural language processing (NLP), and depend on how the models comprehend the given text. If successful, this will have many practical application such as virtual assistants and automated customer service.

Task: One way to test the comprehension ability of a model is by question answering (QA). QA-based tasks must respond to queries presented in natural language based on a context paragraph that contains the answer to the query. In this project, our goal is to built two neural network QA models mainly based on two papers and compare their performances .

Data + Example

SQuAD: Reading comprehension dataset contains 100,000+ question-answer pairs on 500+ articles in a set of Wikipedia articles. Each example has a context, question and 3 reference answers. SQuAD additionaly provides a public competition leaderboard on Stanford-qa.com .





Paragraph

Denver linebacker Von Miller was named Super Bowl MVP, recording five solo tackles, 2½ sacks, and two forced fumbles.

Question

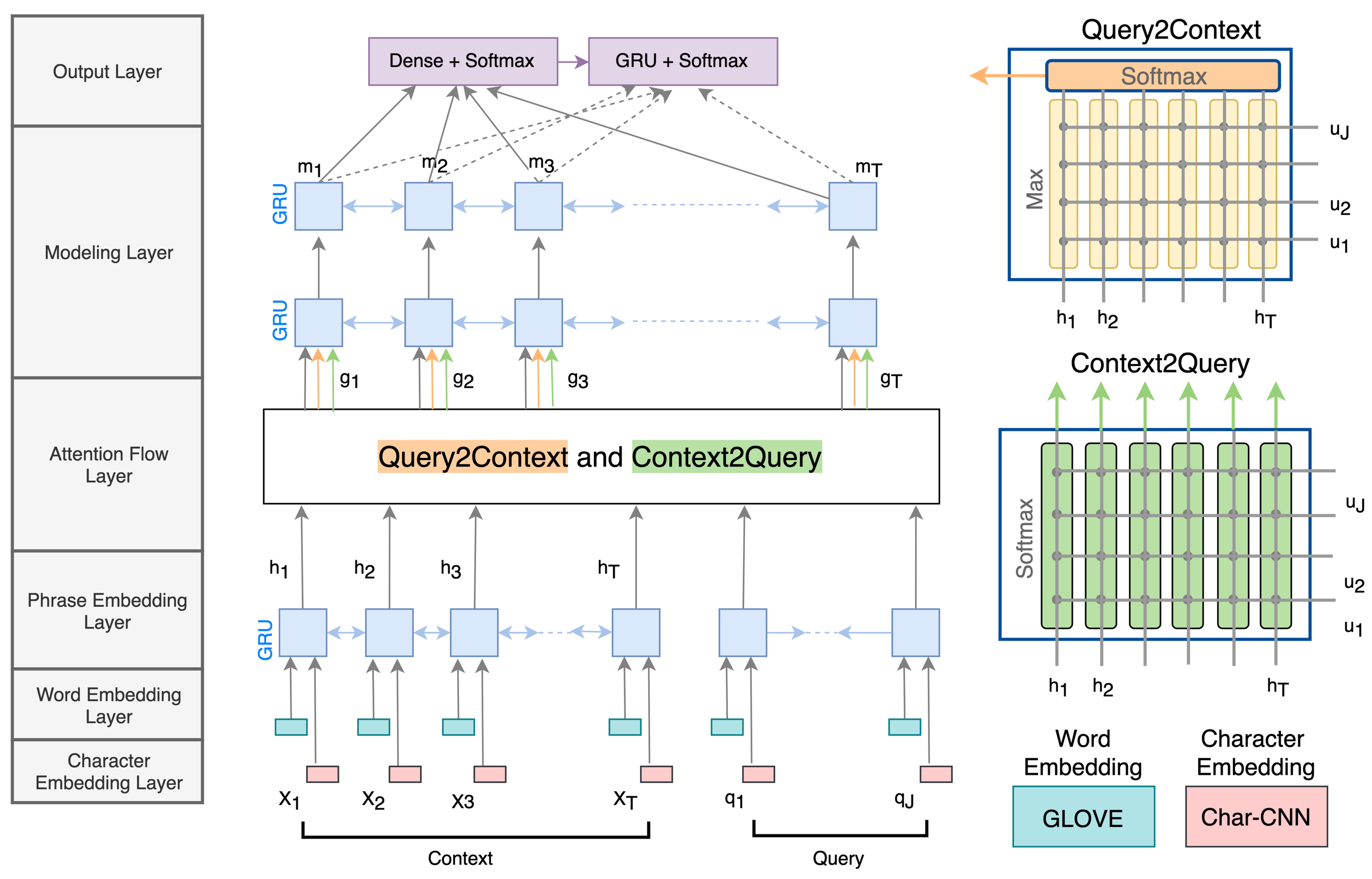
Who was the Super Bowl 50 MVP?

Answer Candidate

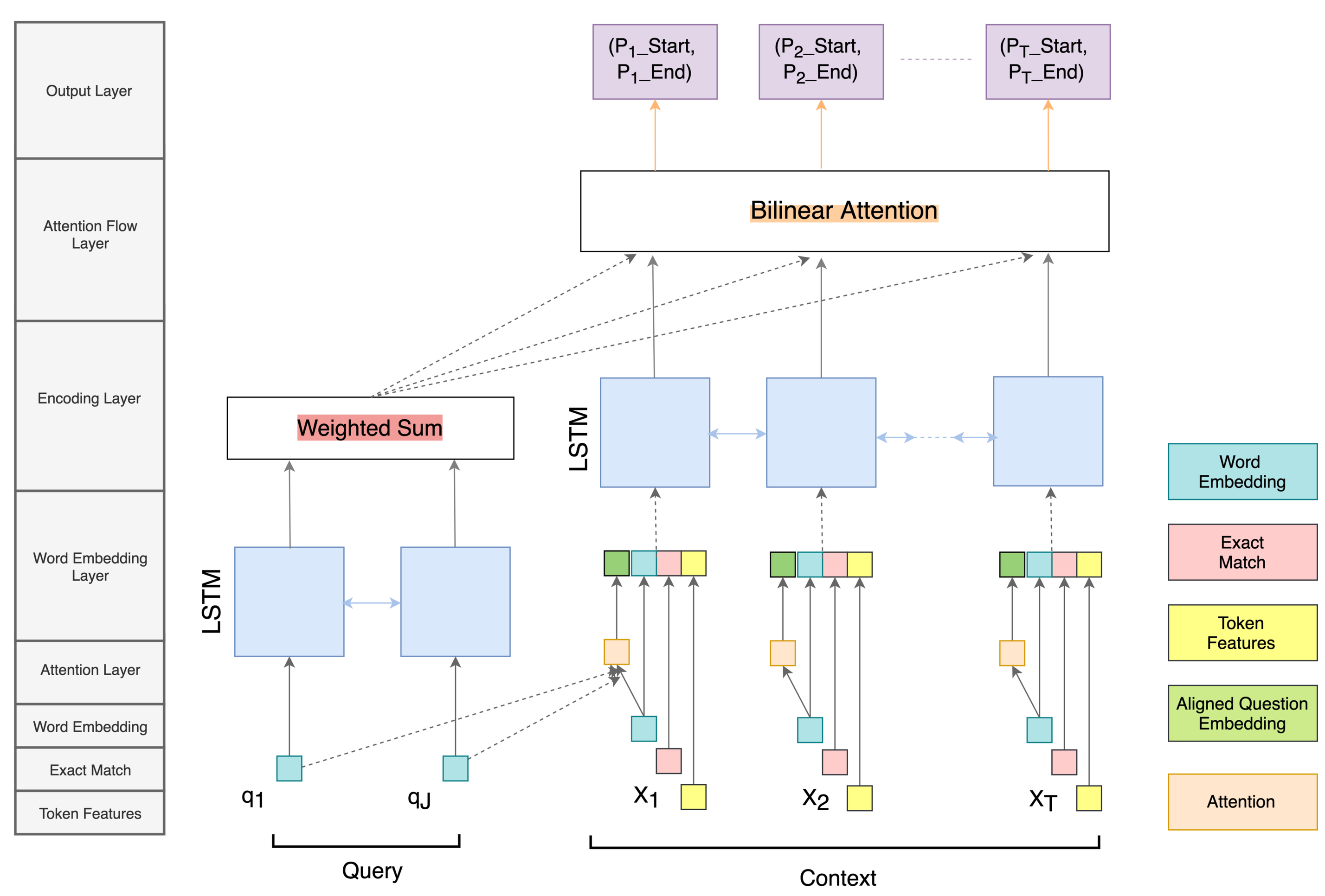
Von Miller

Approaches

Bi-Directional Attention Flow (BiDAF)



Document Reader Question Answering (DrQA)



Embeddings: maps each word to a vector space

- **Character – level :** uses character-level CNNs
- **Word – level :** uses a pre-trained vectors GloVE
- **Contextual – level :** uses contextual cues from surrounding words to refine the embeddings

Attention Flow :

- **Context-to-query:** signifies which query words are most relevant to each context word
- **Query-to-context:** signifies which context words have the closest similarity to one of the query words and are critical for answering the query

Modeling: employs a Recurrent Neural Network to scan the context

Output: provides an answer to the query

Paragraph encoding: represents all tokens p_i in a paragraph as a sequence of feature vectors $\tilde{p}_i \in \mathbb{R}^d$ and pass them as the input to a recurrent neural network and thus obtain:

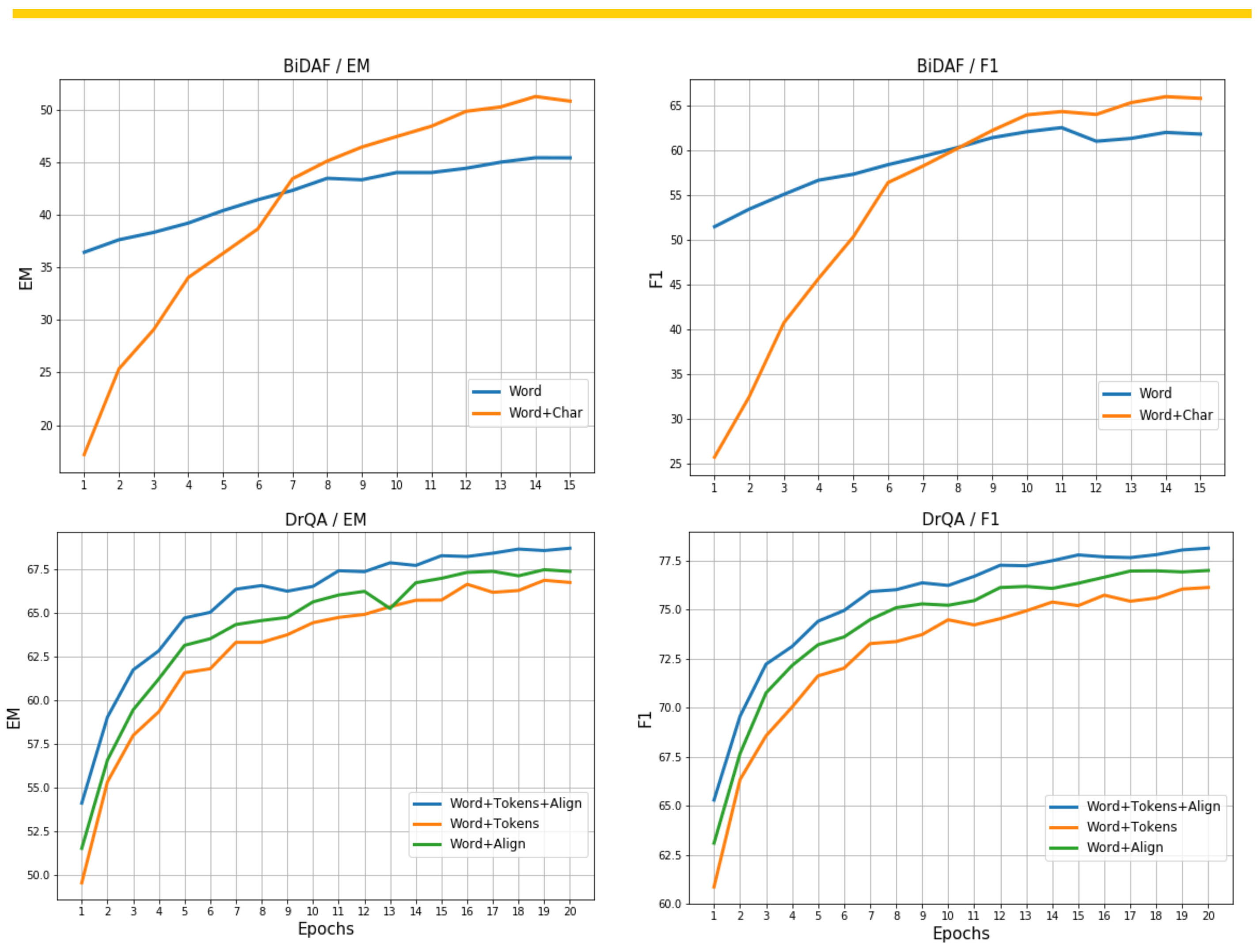
$$\{p_1, \dots, p_m\} = RNN(\{\tilde{p}_1, \dots \tilde{p}_m\})$$

$\tilde{p}_1$ is comprised of the following parts:

- **Word embeddings:** $f_{emb}(p_i) = E(p_i)$
- **Exact match:** $f_{exactmatch}(p_i) = \mathbb{I}(p_i \in q)$
- **Token features:** $$f_{token}(p_i) = (POS(p_i), NER(p_i), TF(p_i))$$
- **Aligned question embedding:** $$f_{align}(p_i) = \Sigma_j(a_{i,j})E(q_i)$$

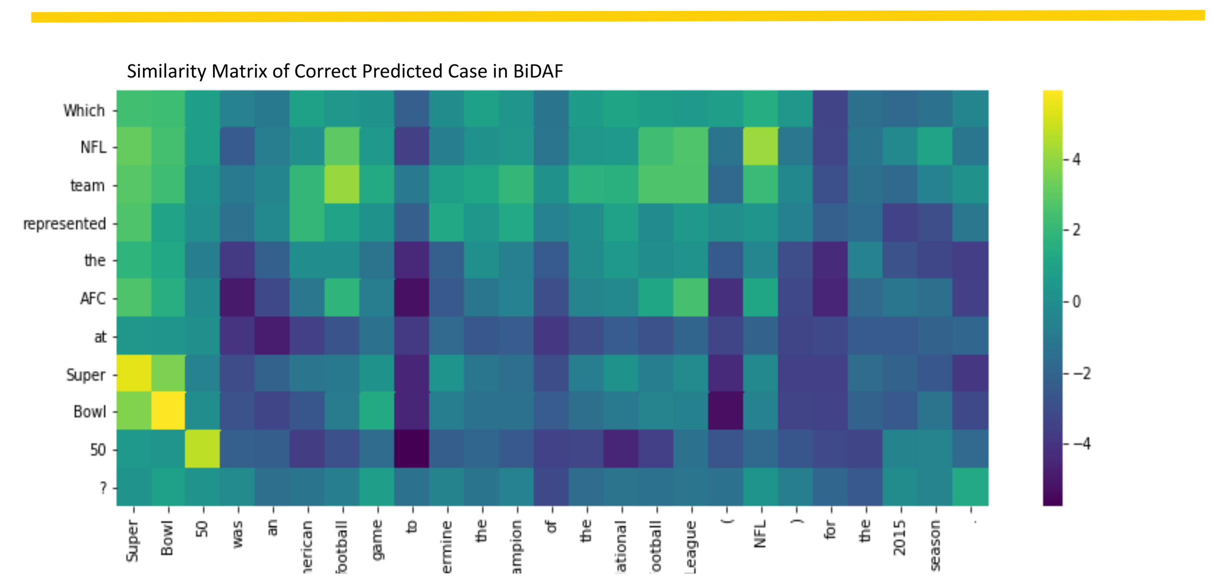
Question encoding: applies another recurrent neural network on top of the word embeddings of q_i and combine the resulting hidden units into one single vector: $\{q_1, \dots, q_l\} \to q$

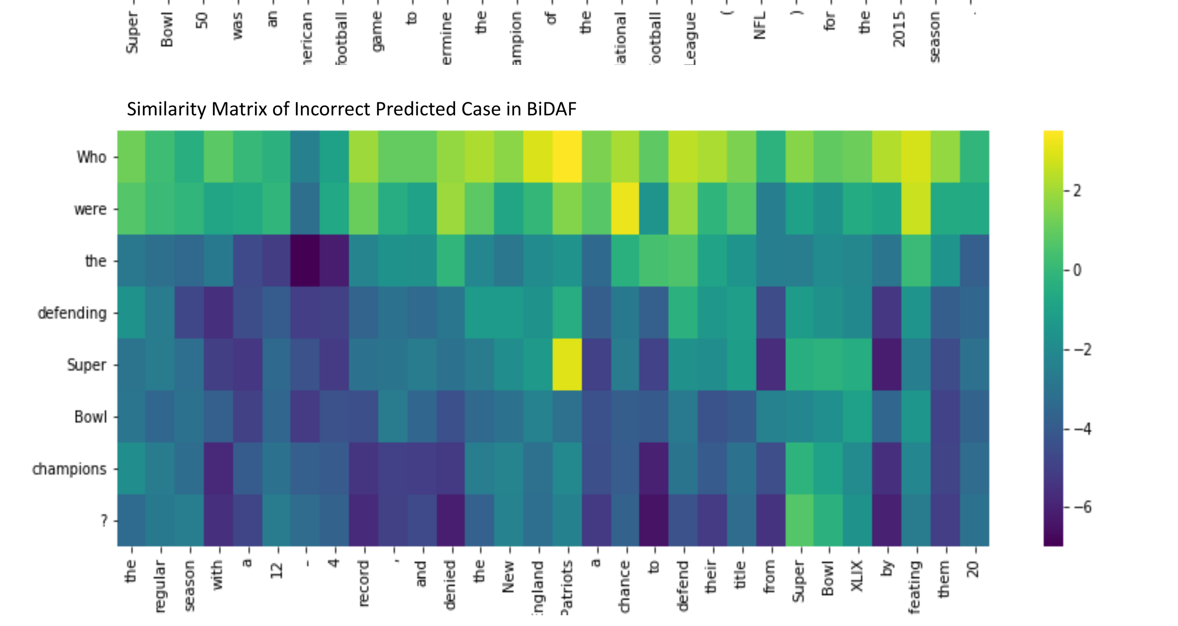
Results



Model	EM	F1
BiDAF + word	45.45	61.84
BiDAF + word + char(CNN)	50.84	65.84
DrQA + word + Align	67.15	76.93
DrQA + word + Tokens	66.77	76.14
DrQA + word + Tokens+ Align	**68.73**	**78.14**

Analysis





- For BiDAF, more complicated embeddings lead to better performances.
- More types of features we add in embedding step, the better performance DrQA has.
- Aligned question embedding works better than token features individually.
- Currently, hyperparameters in BiDAF may not be appropriate. Future work aims to search for optimal ones to make it converge globally.

Reference

- Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, Hannaneh Hajishirzi, *Bidirectional Attention Flow for Machine Comprehension,* arXiv:1611.01603
- Danqi Chen, Adam Fisch, Jason Weston,Antoine Bordes, *Reading Wikipedia to Answer Open-Domain Questions,* arXiv:1704.00051
- Xiang Zhang, Junbo Zhao, Yann LeCun, *Character-level Convolutional Networks for Text Classification,* arXiv:1509.01626
- Rupesh Kumar Srivastava, Klaus Greff, Jürgen Schmidhuber, *Highway Networks,* arXiv:1505.00387