Community Question Answering : Project Report

Damodar Pai Dukle
20162055

Problem Statement :

To Develop a Deep Learning Model for Duplicate Question Detection in Community Question Answering by incorporating and enhancing the traditional Siamese Architecture.

Data Set :

For this task I have made use of Quora Dataset of Duplicate question pairs. The dataset consists of over 400,000 lines of potential question duplicate pairs. Each line contains IDs for each question in the pair, the full text for each question, and a binary value that indicates whether the line truly contains a duplicate pair. There are 255045 negative (non-duplicate) and 149306 positive (duplicate) instances. Making the Total size as 404351.

Motivations :

In this project I have focused on ways to enhance the traditional Siamese Network to incorporate additional information there by making it simpler to detect duplicate question answer pairs.

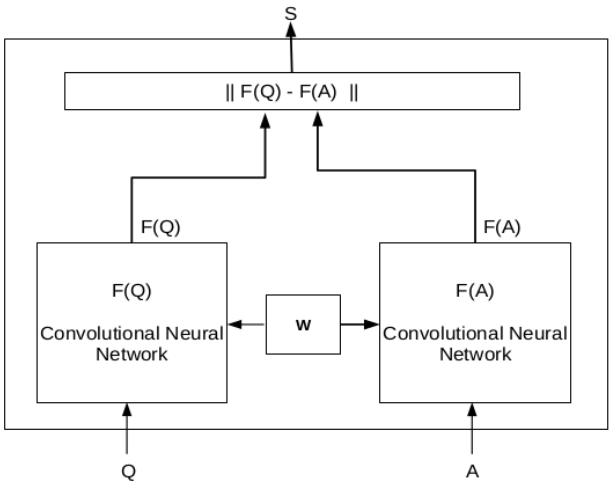
The tradition Siamese Network consists of a Convolution Neural Network with weight sharing between the questions followed by a simple distance metric like Euclidean distance or Cosine similarity.
It uses a contrastive loss function so that semantically similar words are closer in the new vector space than different ones.

Possible ways to enhance the network are as follows :

- Being a CNN based Architecture the Siamese network accepts as inputs character n-gram representation of the words in the questions. Thus every word from the given question is hashed to a character n-gram vocabulary and then passed to the CNN, this while reducing the vocabulary size destroy the semantically sound entity of the word. Thus using Glove/Word2Vec word embeddings should increase semantic coherence of the embedding.
- Using Bag of Character n-grams representation destroys all sequence information thus one needs to investigate sequence information preserving models like LSTMs / GRUs to get better embeddings
- Currently attention based models are doing quite well in solving NLP problems, thus an attempt has been made to add attention in the Siamese network.

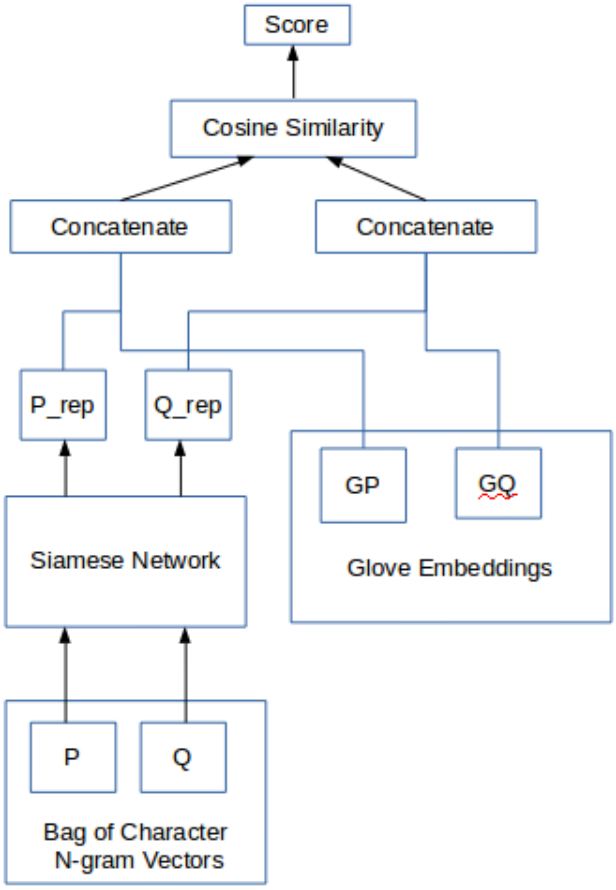
Model Descriptions :

Baseline : Simple Siamese Model



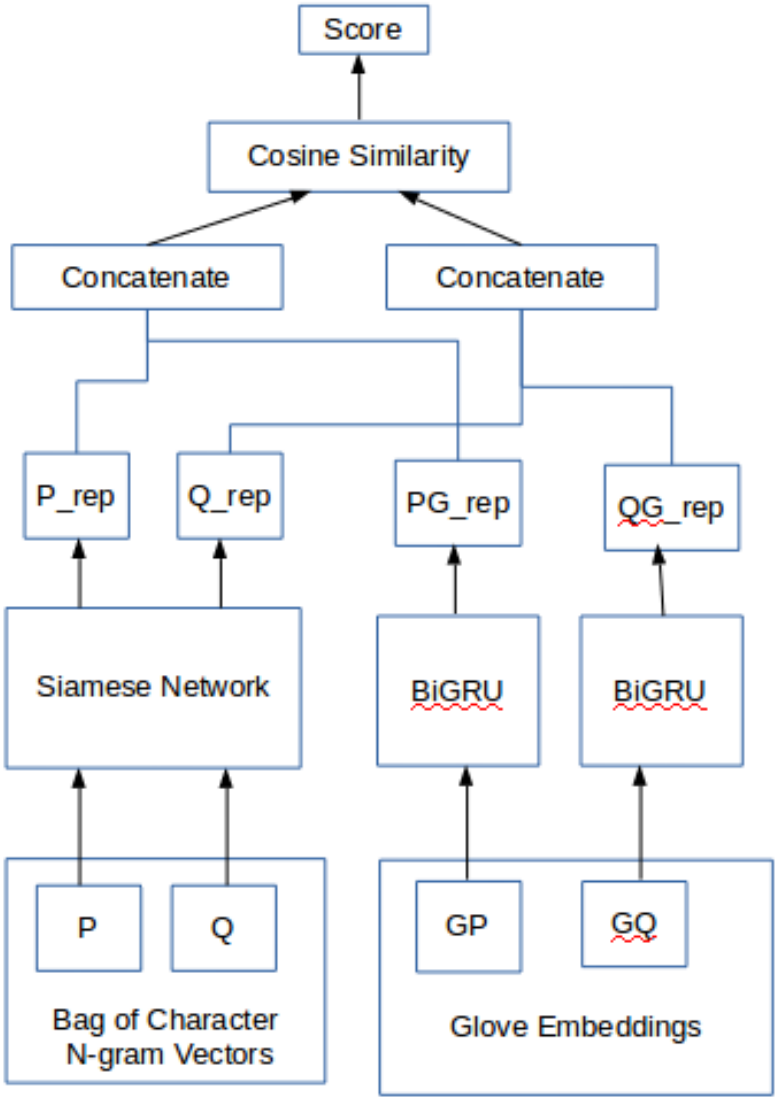
- Hash both questions to character n-gram vocabulary
- Get Embeddings for both questions using the same model (to ensure weight sharing)
- Apply Cosine similarity to generate score / label for similarity between question

Model 1 : Siamese Model with Glove Embeddings



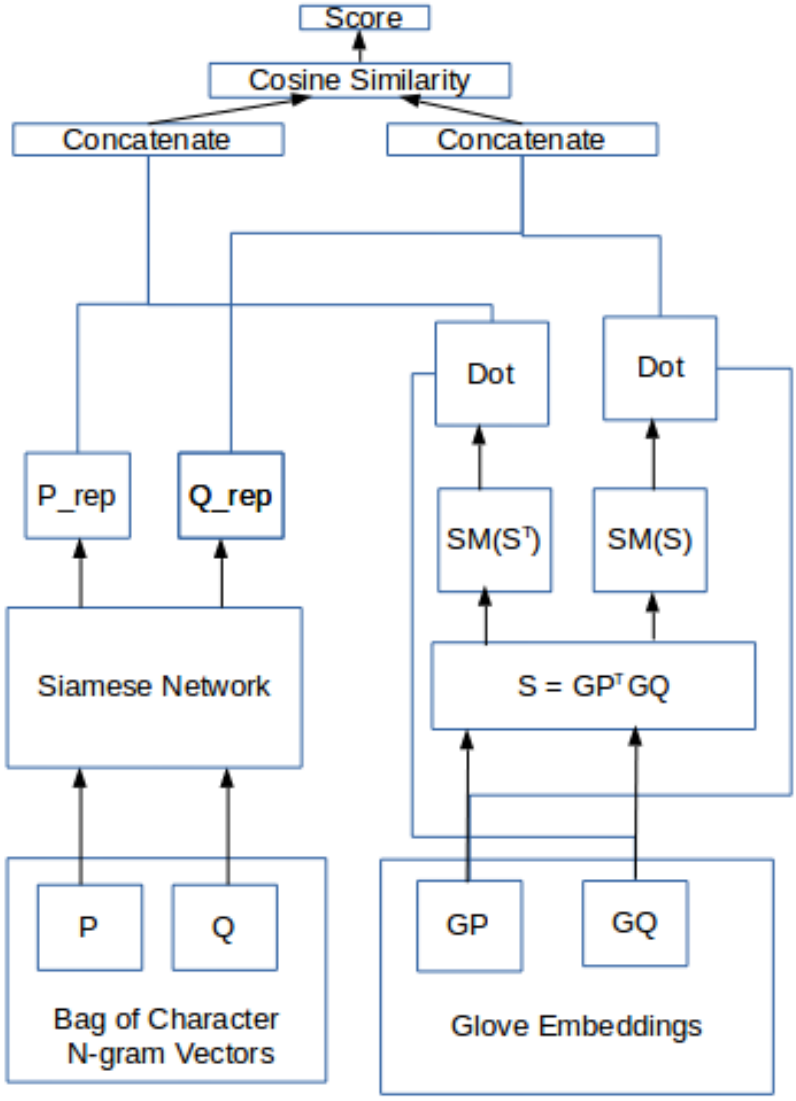
- We Concatenate Siamese embeddings for both questions with their corresponding Glove embedding and feed it to the metric to compute the score

Model 2 : Siamese Model with Sentence Embeddings generated via GRU



- Use Bidirectional GRU units to generate a sentence representation for both the questions from the word wise glove embeddings .
- Concatenate the respective question GRU embeddings with the Siamese embeddings to get the final representation.

Model 3 : Similarity based attention in Siamese Model



- Attention mechanism adopted here is based on word similarity between the questions.
- We compute a similarity matrix (S) then Softmax(SM) is applied row wise to get Attention vector for P and column wise to get attention vector for Q.
- We multiply them with inputs glove embeddings GP and GQ to get contexts for both questions
- These are concatenated and used for scoring

Outputs and Results:

Number of Epochs = 20
Training Data size = 99800
Validation Data size = 100
Test Data size = 100

Baseline system – Simple Siamese Model

```
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            (None, 1, 11522, 1)  0
__________________________________________________________________________________________________
input_2 (InputLayer)            (None, 1, 11522, 1)  0
__________________________________________________________________________________________________
sequential_1 (Sequential)       (None, 128)          28039       input_1[0][0]
                                                                 input_2[0][0]
__________________________________________________________________________________________________
lambda_1 (Lambda)               (None, 1)            0           sequential_1[1][0]
                                                                 sequential_1[2][0]
==================================================================================================
Total params: 28,039
Trainable params: 28,039
Non-trainable params: 0
```

Validation Accuracy : 68% Test Accuracy : 47 %

Model 1 – Baseline with Glove Embeddings

```
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            (None, 1, 11522, 1)  0
__________________________________________________________________________________________________
input_3 (InputLayer)            (None, 20, 300)      0
__________________________________________________________________________________________________
input_2 (InputLayer)            (None, 1, 11522, 1)  0
__________________________________________________________________________________________________
input_4 (InputLayer)            (None, 20, 300)      0
__________________________________________________________________________________________________
sequential_1 (Sequential)       (None, 128)          28039       input_1[0][0]
                                                                 input_2[0][0]
__________________________________________________________________________________________________
reshape_1 (Reshape)             (None, 6000)         0           input_3[0][0]
__________________________________________________________________________________________________
reshape_2 (Reshape)             (None, 6000)         0           input_4[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 6128)         0           sequential_1[1][0]
                                                                 reshape_1[0][0]
__________________________________________________________________________________________________
concatenate_2 (Concatenate)     (None, 6128)         0           sequential_1[2][0]
                                                                 reshape_2[0][0]
__________________________________________________________________________________________________
lambda_1 (Lambda)               (None, 1)            0           concatenate_1[0][0]
                                                                 concatenate_2[0][0]
==================================================================================================
Total params: 28,039
Trainable params: 28,039
Non-trainable params: 0
__________________________________________________________________________________________________
```

Validation Accuracy : 64% Test Accuracy : 50%

Model 2 – Baseline with Glove and GRU

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 1, 11522, 1)	0	
input_3 (InputLayer)	(None, 20, 300)	0	
input_2 (InputLayer)	(None, 1, 11522, 1)	0	
input_4 (InputLayer)	(None, 20, 300)	0	
sequential_1 (Sequential)	(None, 128)	22919	input_1[0][0] input_2[0][0]
bidirectional_1 (Bidirectional)	(None, 256)	329472	input_3[0][0]
bidirectional_2 (Bidirectional)	(None, 256)	329472	input_4[0][0]
concatenate_1 (Concatenate)	(None, 384)	0	sequential_1[1][0] bidirectional_1[0][0]
concatenate_2 (Concatenate)	(None, 384)	0	sequential_1[2][0] bidirectional_2[0][0]
lambda_1 (Lambda)	(None, 1)	0	concatenate_1[0][0] concatenate_2[0][0]

Total params: 681,863
Trainable params: 681,863
Non-trainable params: 0

Validation Accuracy : 72% Test Accuracy : 61%

Model 3 – Baseline with Glove and Attention in Siamese Model

Layer (type)	Output Shape	Param #	Connected to
input_3 (InputLayer)	(None, 20, 300)	0	
input_4 (InputLayer)	(None, 20, 300)	0	
dot_1 (Dot)	(None, 20, 20)	0	input_3[0][0] input_4[0][0]
permute_1 (Permute)	(None, 20, 20)	0	dot_1[0][0]
activation_3 (Activation)	(None, 20, 20)	0	permute_1[0][0]
activation_4 (Activation)	(None, 20, 20)	0	dot_1[0][0]
input_1 (InputLayer)	(None, 1, 11522, 1)	0	
dot_2 (Dot)	(None, 20, 300)	0	activation_3[0][0] input_4[0][0]
input_2 (InputLayer)	(None, 1, 11522, 1)	0	
dot_3 (Dot)	(None, 20, 300)	0	activation_4[0][0] input_3[0][0]
sequential_1 (Sequential)	(None, 128)	28039	input_1[0][0] input_2[0][0]
reshape_1 (Reshape)	(None, 6000)	0	dot_2[0][0]
reshape_2 (Reshape)	(None, 6000)	0	dot_3[0][0]
concatenate_1 (Concatenate)	(None, 6128)	0	sequential_1[1][0] reshape_1[0][0]
concatenate_2 (Concatenate)	(None, 6128)	0	sequential_1[2][0] reshape_2[0][0]
lambda_1 (Lambda)	(None, 1)	0	concatenate_1[0][0] concatenate_2[0][0]

Total params: 28,039
Trainable params: 28,039
Non-trainable params: 0

Validation Accuracy : 63% Test Accuracy : **68 %**

Model Comparison:

P	Q	Actual Label	Baseline Prediction	Model 1 Prediction	Model 2 Prediction	Model 3 Prediction
Which books and magazines should an MBA student read?	What are the books that can mould a mba student towards bright future ?	1	False,0.28077486	False,0.025833435	False,0.08094262	False,0.4064989
Can you view a private Facebook profile? How can you do this?	How do you view a private Facebook profile?	1	False,-0.14906542	False,-0.028812457	False,0.17504671	True,0.8556741
Who's winning the election, Trump or Clinton?	Who will win the Election? Trump or Clinton?	1	False,0.1897493	False,0.42839134	True,0.97168756	True,0.8457537
Is providing family resource information services from a website considered Plagiarism?	Is providing family resource information services considered Plagiarism?	1	False,0.30775982	False,0.44504243	True,0.81753325	False,0.34467968
How can I hack the others Facebook account?	What are some ways to hack a Facebook account?	1	False,0.4392959	True,0.92189705	True,0.62133896	False,-0.20280309
Who would win: Black Panther or Batman?	Who would win in a fight between Black Panther and Batman?	1	False,0.08139222	True,0.8564042	False,0.2148472	True,0.8659599
Can I find or track my lost mobile device using the phone number?	How do I track someone from his mobile number?	1	False,0.08878588	True,0.9716619	False,0.33027536	False,0.44018245
If Barack Obama ran against Donald Trump who would win the presidential election?	Hypothetical Scenarios: Who would win the US election - Barack Obama (2008) vs Donald Trump (2016)?	1	True,0.6330671 3	True,0.5666140 3	True,0.5194480 4	True,0.8659568
What are the health benefits of herbal tea?	What are the benefits of drinking natural herbal tea?	1	True,0.8554764	True,0.6171981 7	True,0.9879945 5	False,0.3013922
What will Google name their Android versions after they finish with the alphabet "Z"?	What will the name of the future versions of Android be after the last Z word is used?	1	True,0.6720655	True,0.6528127	False,0.17115143	True,0.6864563
How QuickBooks Proavisor Tech support Phone Number is Prominent for getting Solutions?	What is Quickbooks tech support number in Arizona?	1	True,0.5364358 4	True,0.5143391	False,0.16096917	False,0.028177619
Should I allow my 15-year-old daughter to have a sleep over with a friend who's a boy?	Should I allow my 15-year-old daughter to have a sleep over with her boyfriend?	1	True,0.7523644	False,0.4472049	True,0.5676602	True,0.7619133
Which programming language should a beginner learn first?	Which is the best programming language to learn for hacking? What are some books for beginners?	1	True,0.4989300 4	False,0.4717707 6	True,0.7864529 5	False,0.4281918
How can you stop caring about someone who doesn't care about you?	How can I stop caring about a girl who doesn't care about me?	1	True,0.7726123 3	False,0.1836850 5	False,0.2856679 6	False,-0.0017379355

Conclusion

- Both GRU based and Attention based model add valuable new information to the Siamese architecture.
- Attention based Siamese works well to capture semantic information of question along with the n-gram similarity captured by traditional network.
- Using learning rate decay and Relu activation while scoring results in better results