Identifying Duplicate Question Pairs

Aditya Garg(14033)
Bhargav Ganguly(14177)
Gaurav Kumar(14240)
Ankit Singh(14106)
Shubham Jain(14676)

IIT Kanpur

April 18, 2018

Problem Statement

- Given two different sentences, aim is to predict the degree of semantic similarity between them.
- The objective can be defined mathematically in the following manner. Given two question pairs $q1$ and $q2$, learn a model $\mathcal{M}$ such that
  - $\mathcal{M}(q1, q2) = 1$ if q1 & q2 have semantic similarity
  - $\mathcal{M}(q1, q2) = 0$ otherwise
- Model returns the probability of relatedness which would be thresholded to get the final binary output.

Dataset

- For this problem, we would be using the labeled Quora Question database.
- Here is a snippet of the dataset.

id	qid1	qid2	question1	question2	is_duplicate
447	895	896	What are natural numbers?	What is a least natural number?	0
1518	3037	3038	Which pizzas are the most popularly ordered pizzas on Domino's menu?	How many calories does a Dominos pizza have?	0
3272	6542	6543	How do you start a bakery?	How can one start a bakery business?	1
3362	6722	6723	Should I learn python or Java first?	If I had to choose between learning Java and Python, what should I choose to learn first?	1

- Each training example consists of two questions and their respective question ID's.
- Each example has a binary label 1 or 0 where 1 means that question pair is similar

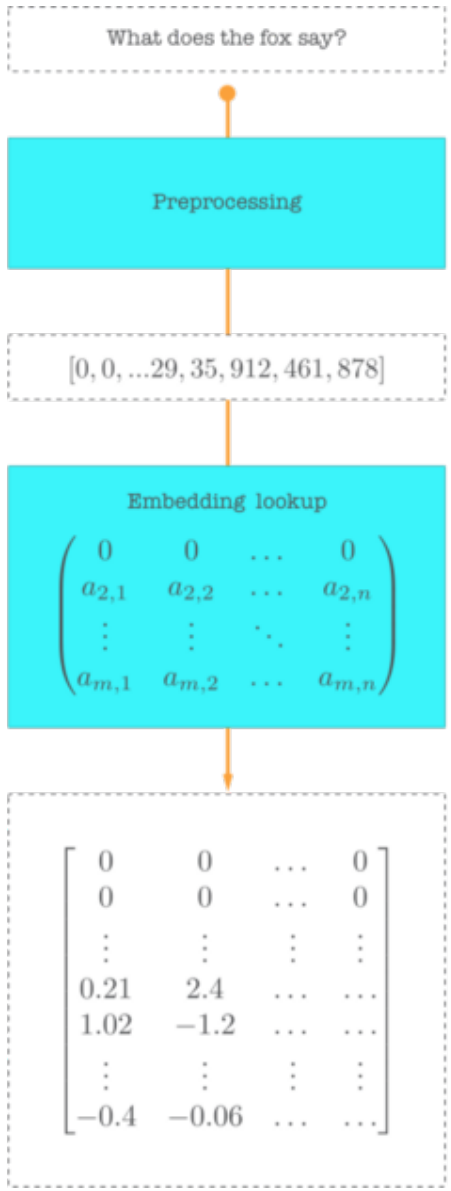
Motivation

- Quora is a platform for asking questions and answering them
- Since, anybody can ask any question, chances of question duplication are high. Same question may be asked by multiple users.
- Important to identify such redundancies and remove them.
- User can be redirected to a similar answered question
- It requires a learning model to perform this task automatically

Pre Processing

- **Tokenization** is done on all the questions
- **Padding** is done for each question to have max sequence length of 25.
- Pretrained **Glove 840B 300d** matrix was used to generate word embeddings for words in the vocabulary.

Diagram showing preprocessing



RAW INPUT

VECTOR OF WORD INDICES REPRESENTING THE QUESTION

MATRIX REPRESENTING THE QUESTION WHERE EACH VECTOR IS THE EMBEDDING OF ITS CORRESPONDING WORD

Approaches used

- LSTM with concatenation
- LSTM with distance and angle
- Siamese LSTM with Manhattan distance
- Max-bag of embedding approach
- Neural bag of words

LSTM with Concatenation

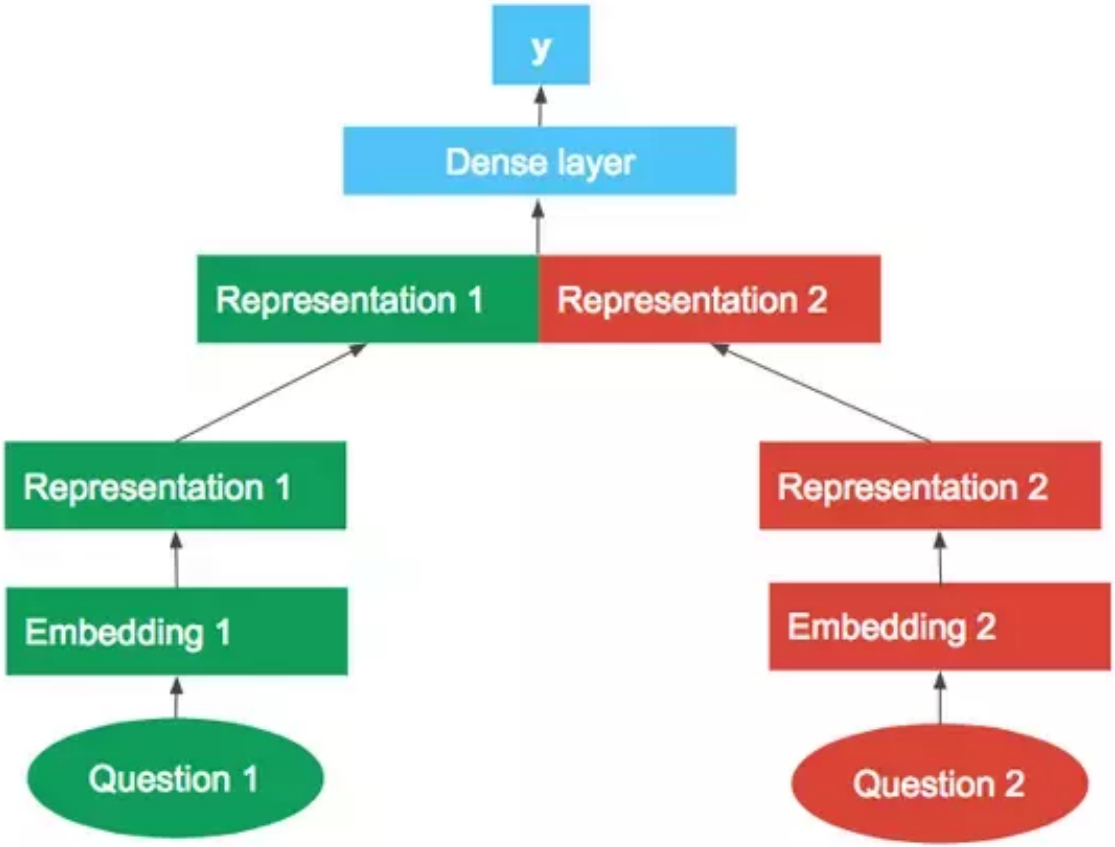


Figure: Visual representation of Approach 1

LSTM with Distance and Angle

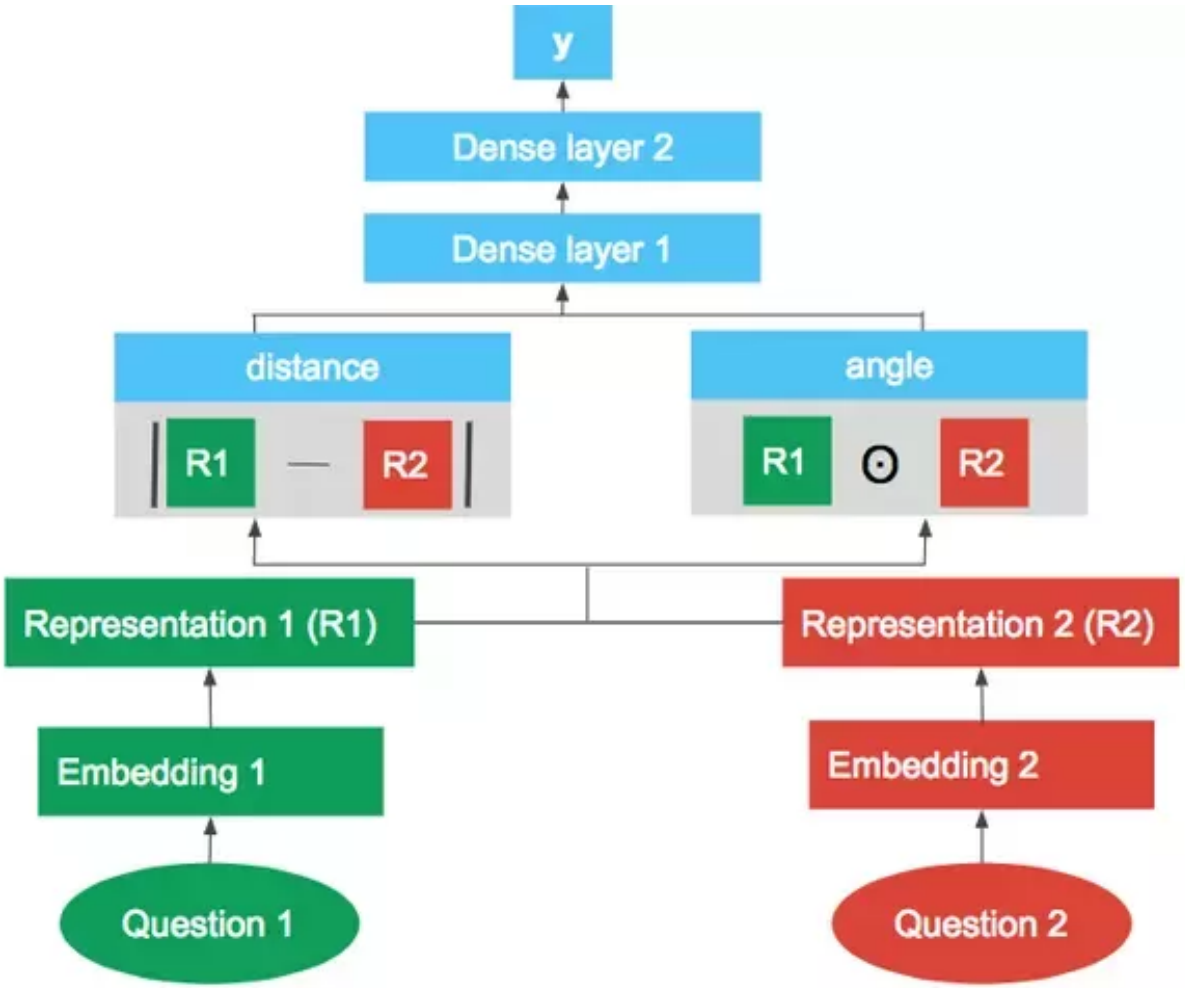


Figure: Visual representation of Approach 2

Siamese LSTM with Manhattan Distance

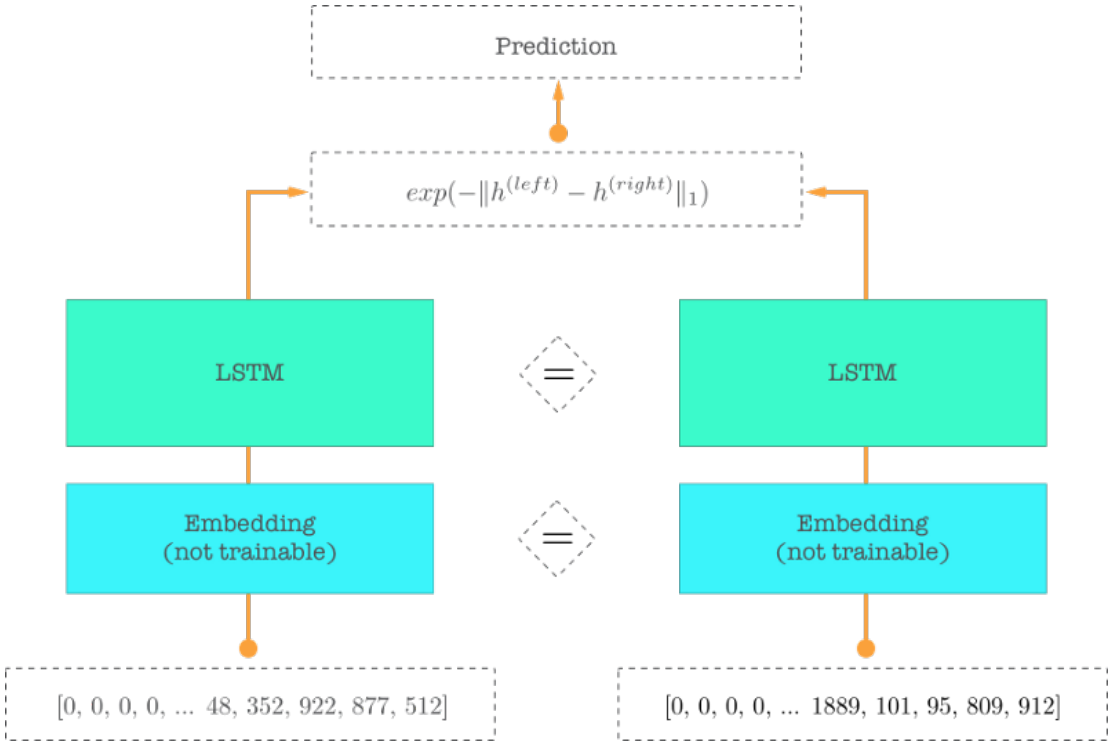
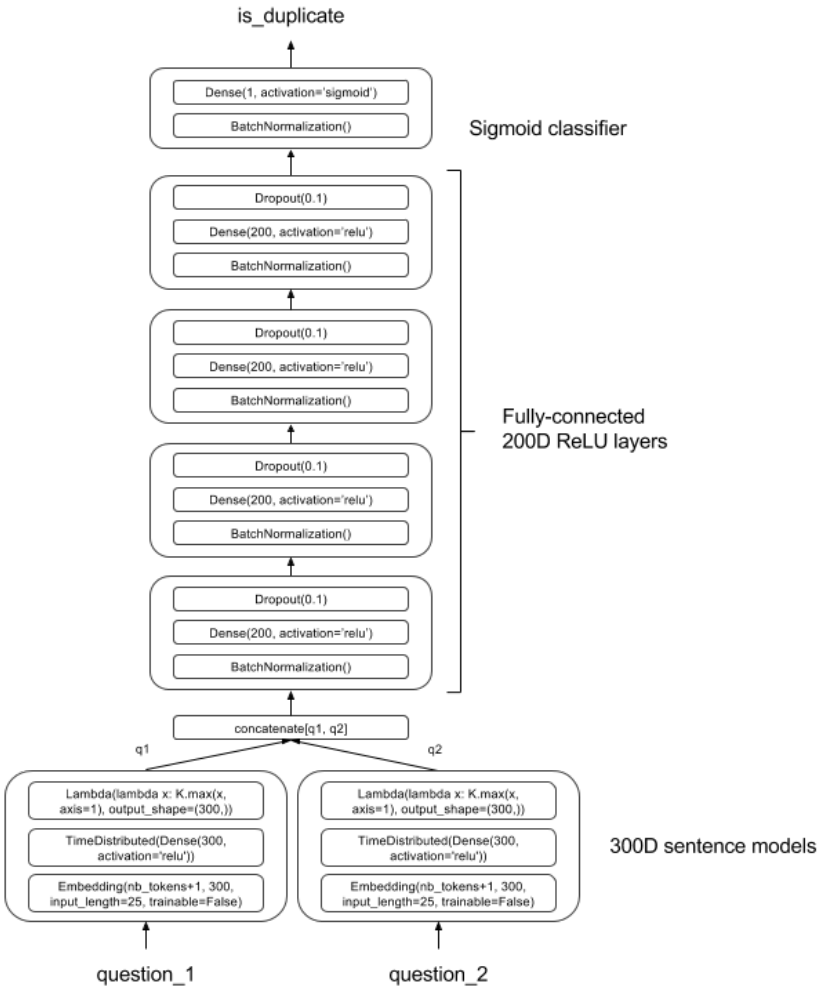
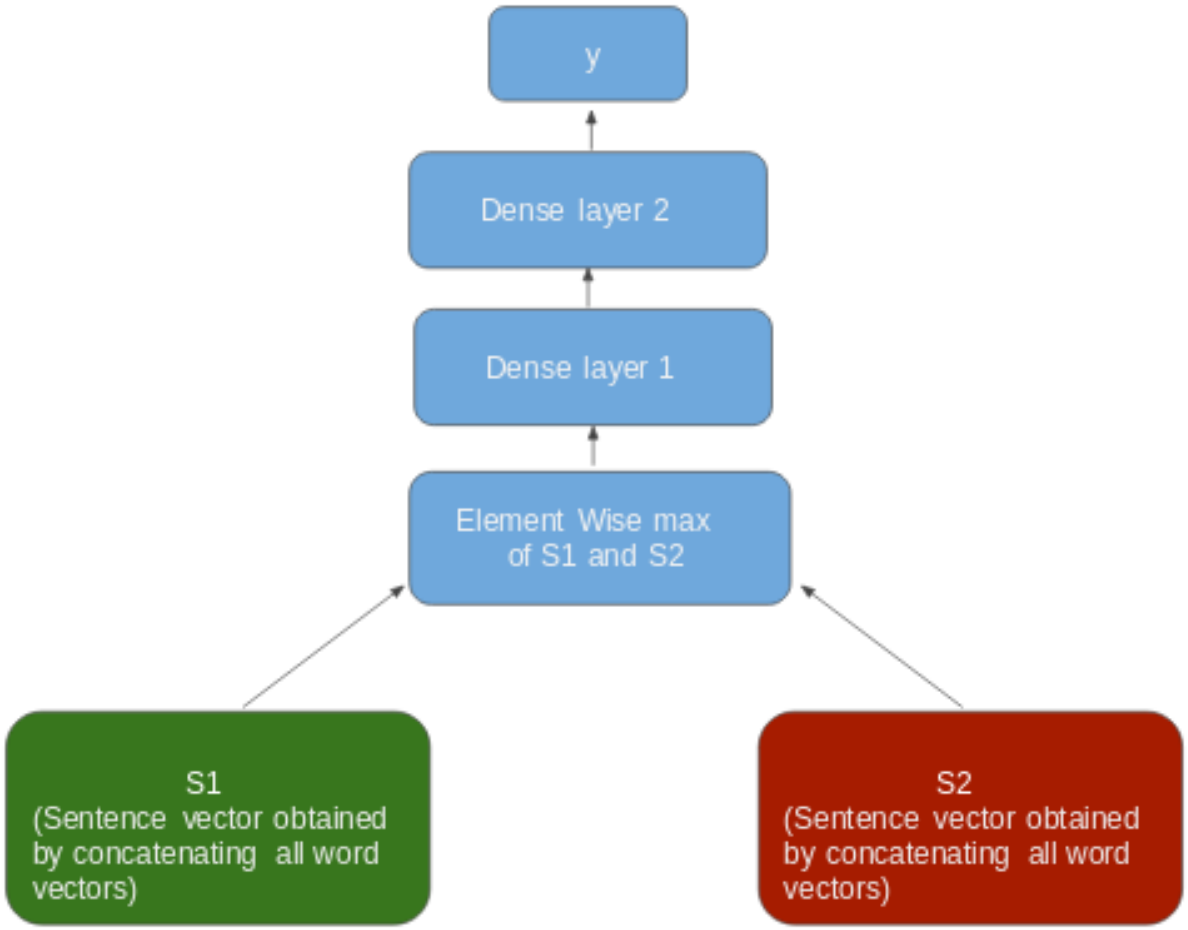


Figure: Word index for each question is fed as inputs

Max Bag of Embeddings



Neural bag of Words



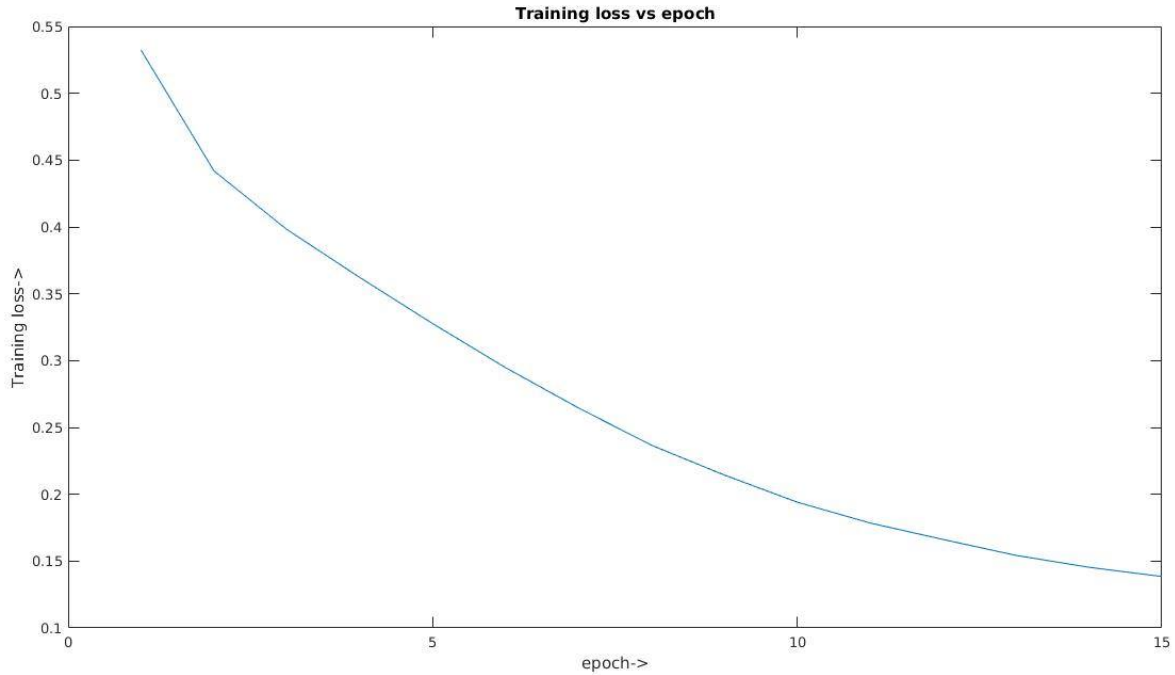
Hyperparameters Tuning

- **Loss Function** : Binary Cross Entropy for all methods
- **Dropout** : 0.1 for all methods
- LSTM with Concatenation
  - **Optimizer**: Adam
  - **Batch Size**: 128
  - **Epochs**: 25
- LSTM with Distance and Angle
  - **Optimizer**: Adam
  - **Batch Size**: 128
  - **Epochs**: 25

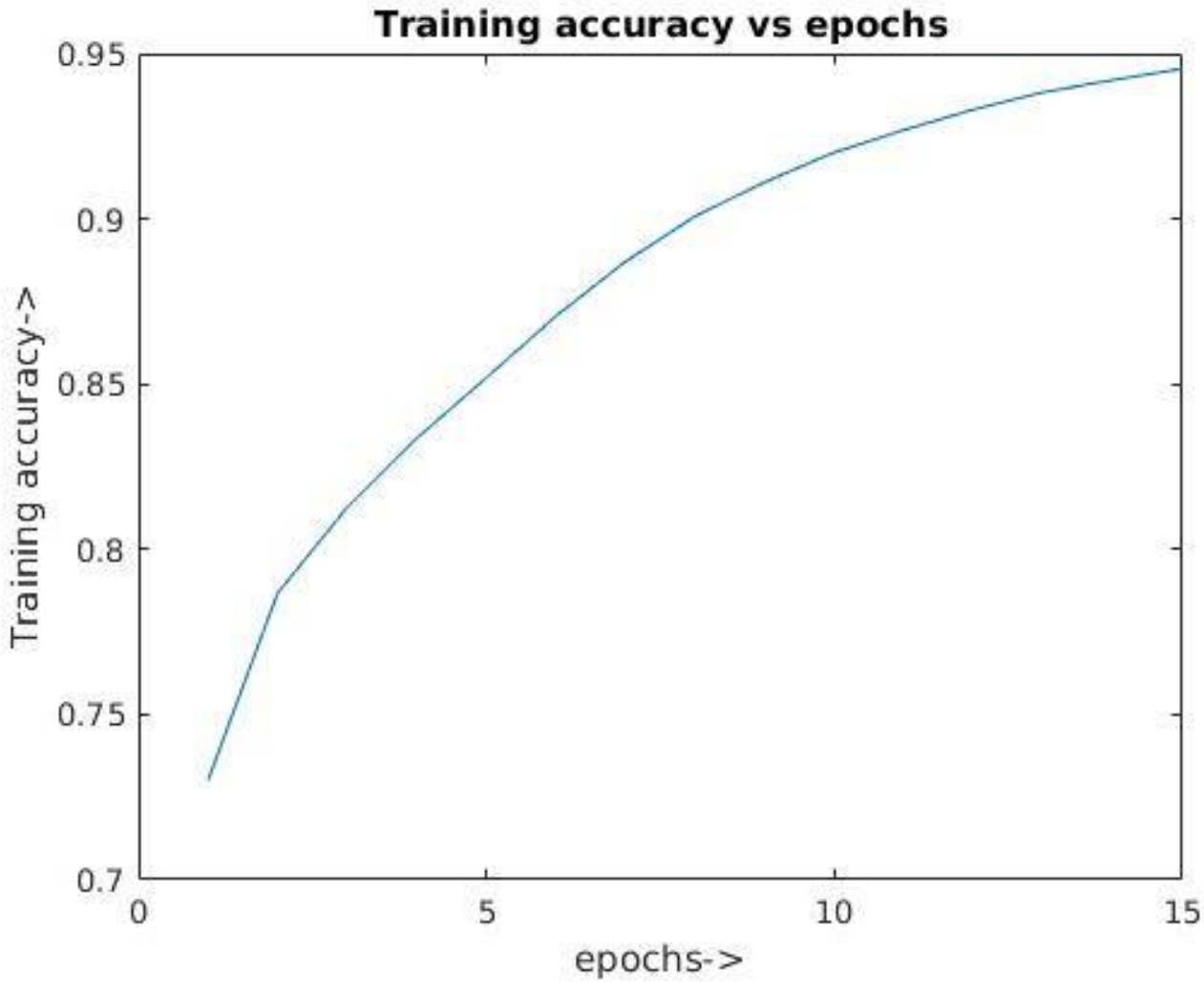
HyperParameter Tuning

- Siamese LSTM with Manhattan distance
  - **Optimizer**: AdaDelta
  - **Batch Size**: 32
  - **Epochs**: 10
  - **Gradient Clipping Norm**: 1.25
- Max bag of Embedding
  - **Optimizer**: Adam
  - **Batch Size**: 128
  - **Epochs**: 25
- Neural Bag of Words
  - **Optimizer**: Adam
  - **Batch Size**: 32
  - **Epochs**: 25

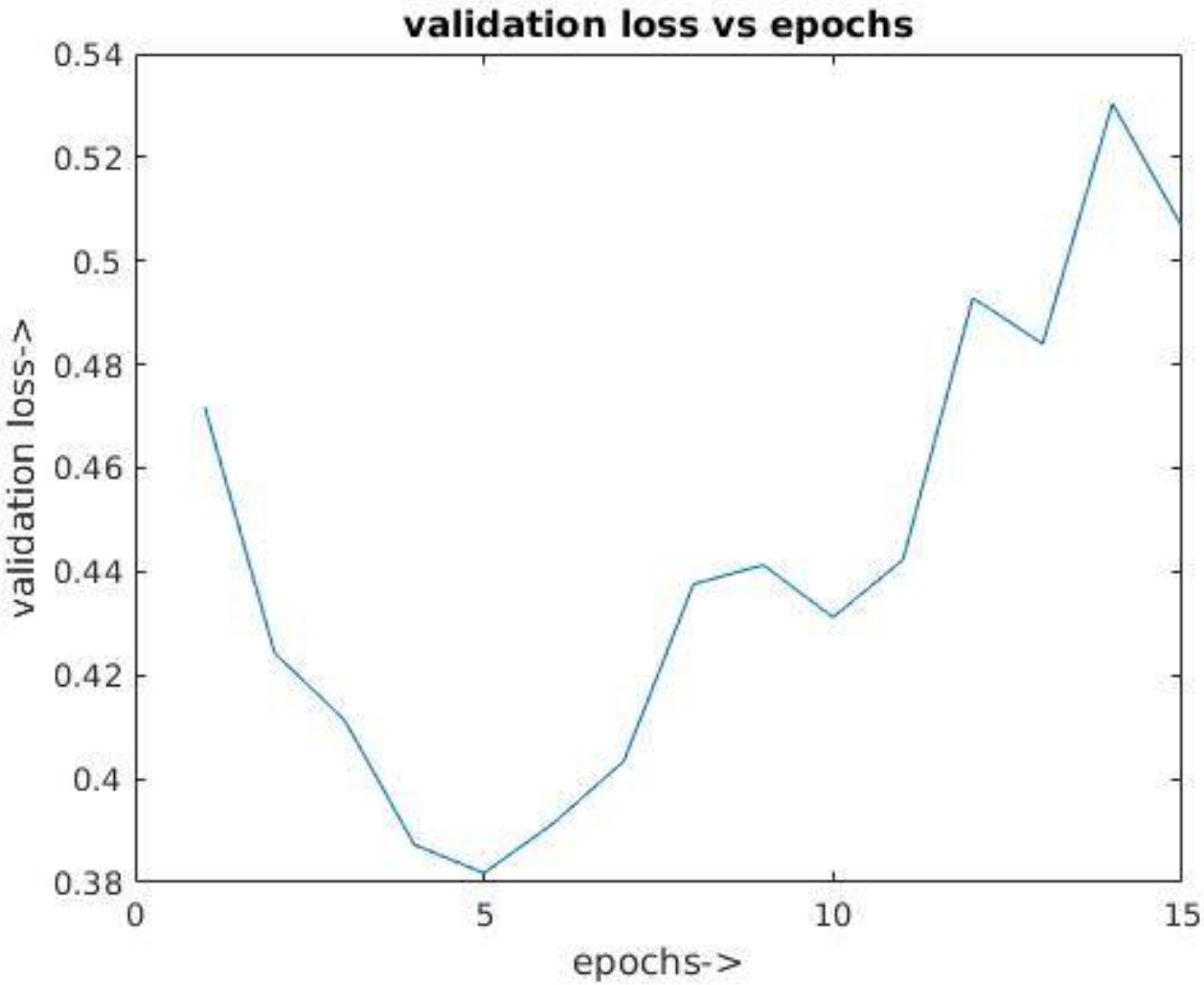
Plot between Training Loss and Epoch



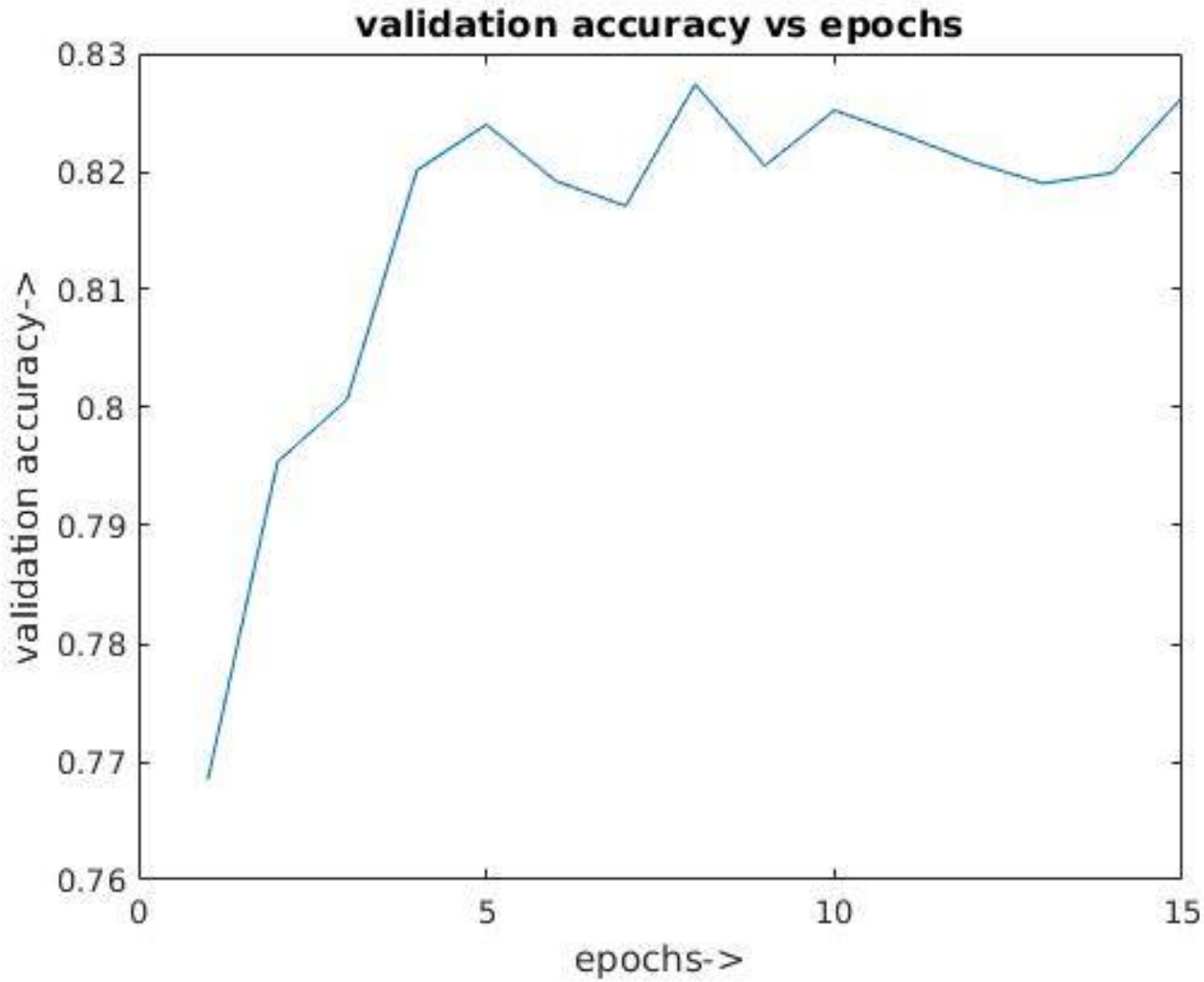
Plot between Training Accuracy and Epoch



Validation Loss vs Epoch



Validation Accuracy vs Epoch



Accuracies

Models	Accuracy
LSTM with concatenation	82.18
LSTM with distance and angle	82.55
Siamese LSTM with Manhattan distance	82.11
Max-bag of embedding approach	81.50
Neural Bag of words	79.84

References

- Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing.
- Ankur P Parikh, Oscar Tckstrm, Dipanjan Das, Jakob Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference, arXiv 2016
- Sepp Hochreiter, Jurgen Schmidhuber, Long Short Term Memory, Neural Computation 1997
- N Srivastava, G Hinton, A Krizhevsky, I Sutskever, R Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, JMLR 2014

The End