Quora questions similarity

Konstantin Lekomtsev
Data Science Career Track

Mentor: Nishant Gupta

Objective and methods, Dataset information

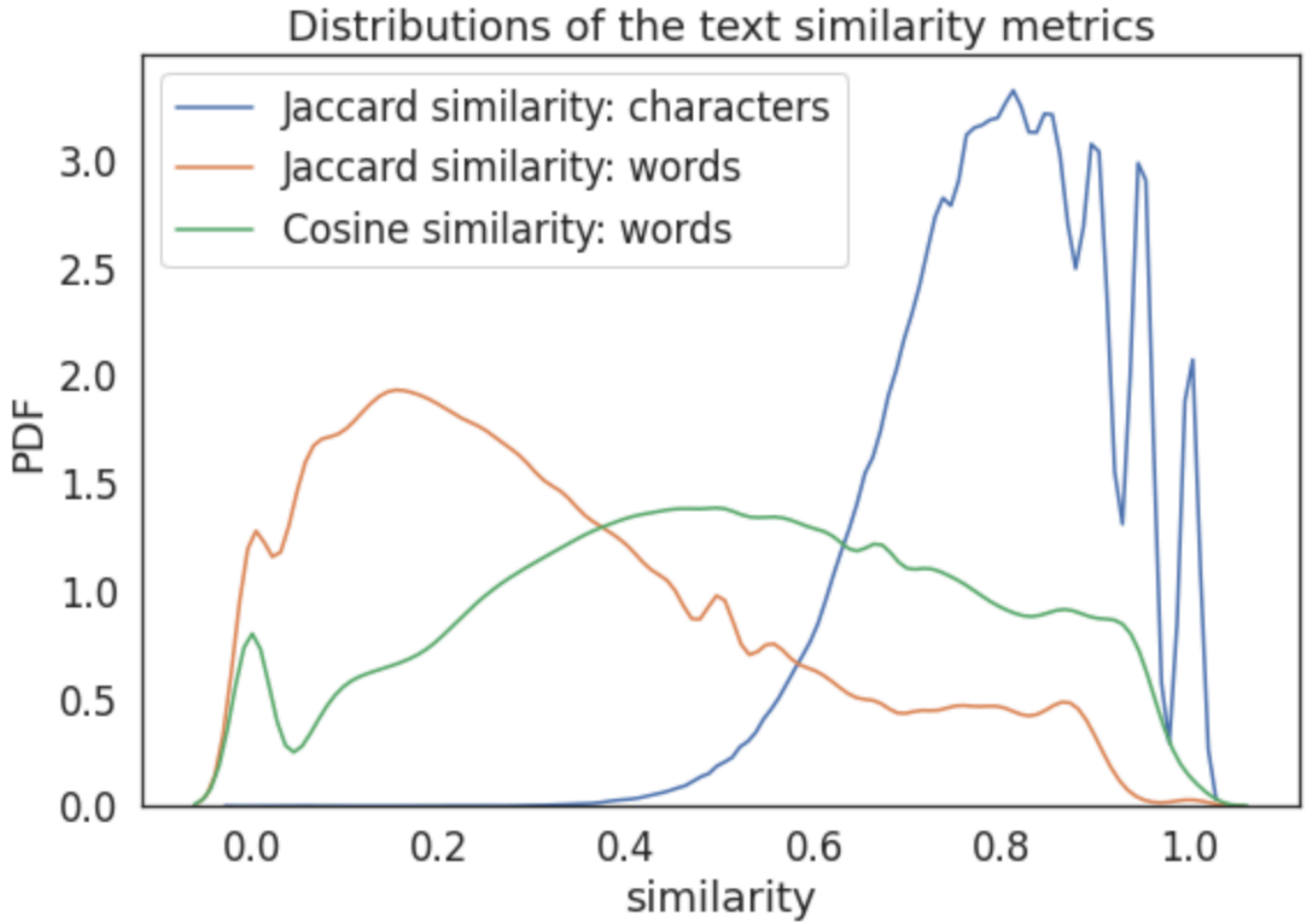
Objective: identify duplicate question pairs.

Methods: one-shot learning using twin neural networks with time distributed and LSTM layers. Questions embeddings were done using pre-trained GloVe embeddings.

Dataset info:
Total number of questions pairs: 404,278
Portion of not duplicate reviews: 0.63
Portion of duplicate reviews: 0.37
Maximum question length, question 1: 125
Maximum question length, question 2: 237
Median number of words per question: 10
Vocabulary size: 232531 words

Similarity metrics

- Jaccard similarity, based on characters and words.
- Cosine similarity between questions, based on words.



*** The Jaccard similarity index compares members for two sets to see which members are shared and which are distinct.**

*** Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them.**

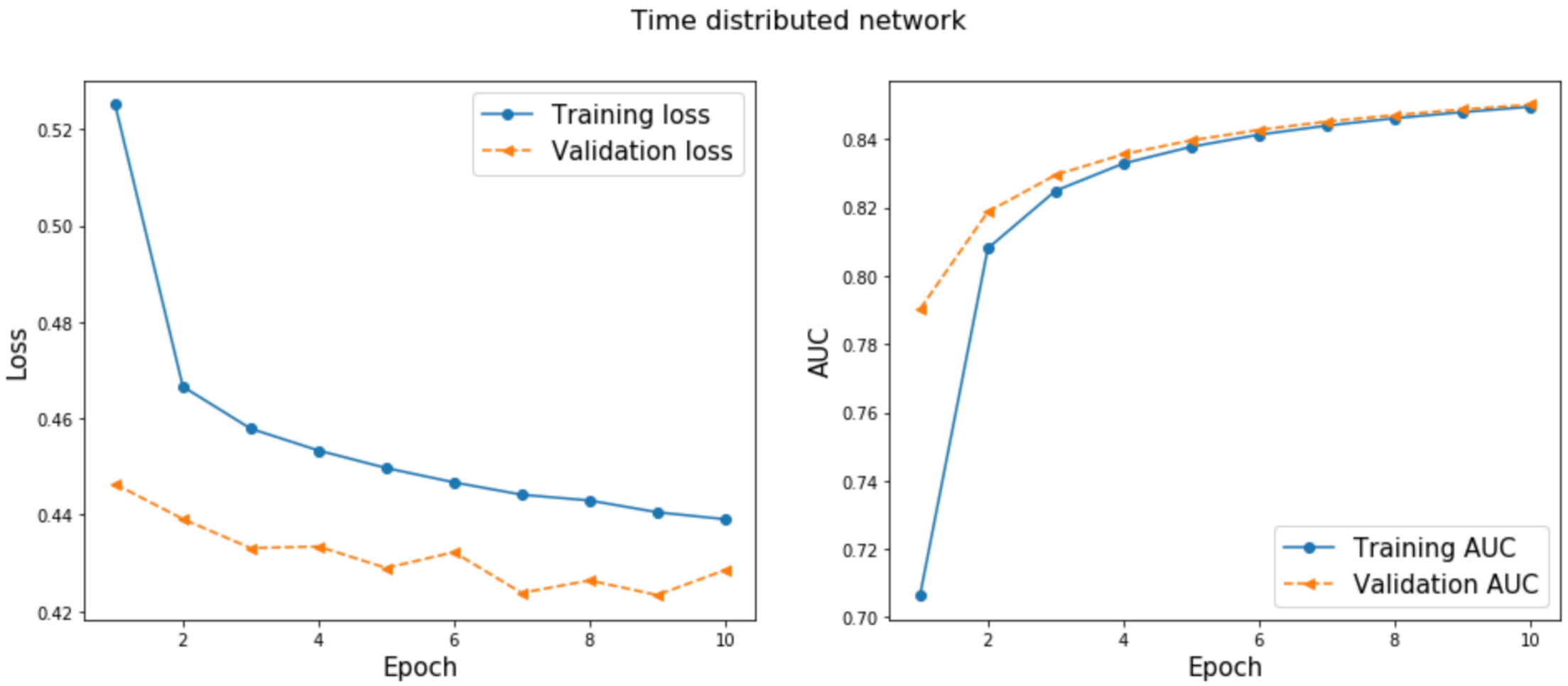
Network with time distributed layer

One-shot learning using a twin neural network with time distributed layers, and cosine and Jaccard similarity.

- Add embedding layers and then time distributed layers for each question.
- Calculate similarity features: cosine similarity on words, Jaccard similarity on characters and words.
- Add features equivalent to finding the most similar words in question pairs (max value along the embedding vector in the tensor for each question) and average similarity of question pairs (sum of values along the dimension in the tensor corresponding to the sequence of words in each question).

*** The time distributed layer applies a dense layer to every temporal slice of an input. The input should be at least 3D, and the dimension of index one will be considered to be the temporal dimension. Consider a batch of 32 samples, where each sample is a sequence of 10 vectors of 16 dimensions. The batch input shape of the layer is then (32, 10, 16), and the input shape, not including the dimension of the sample, is (10, 16).**

Network with time distributed layer - training



Network with LSTM

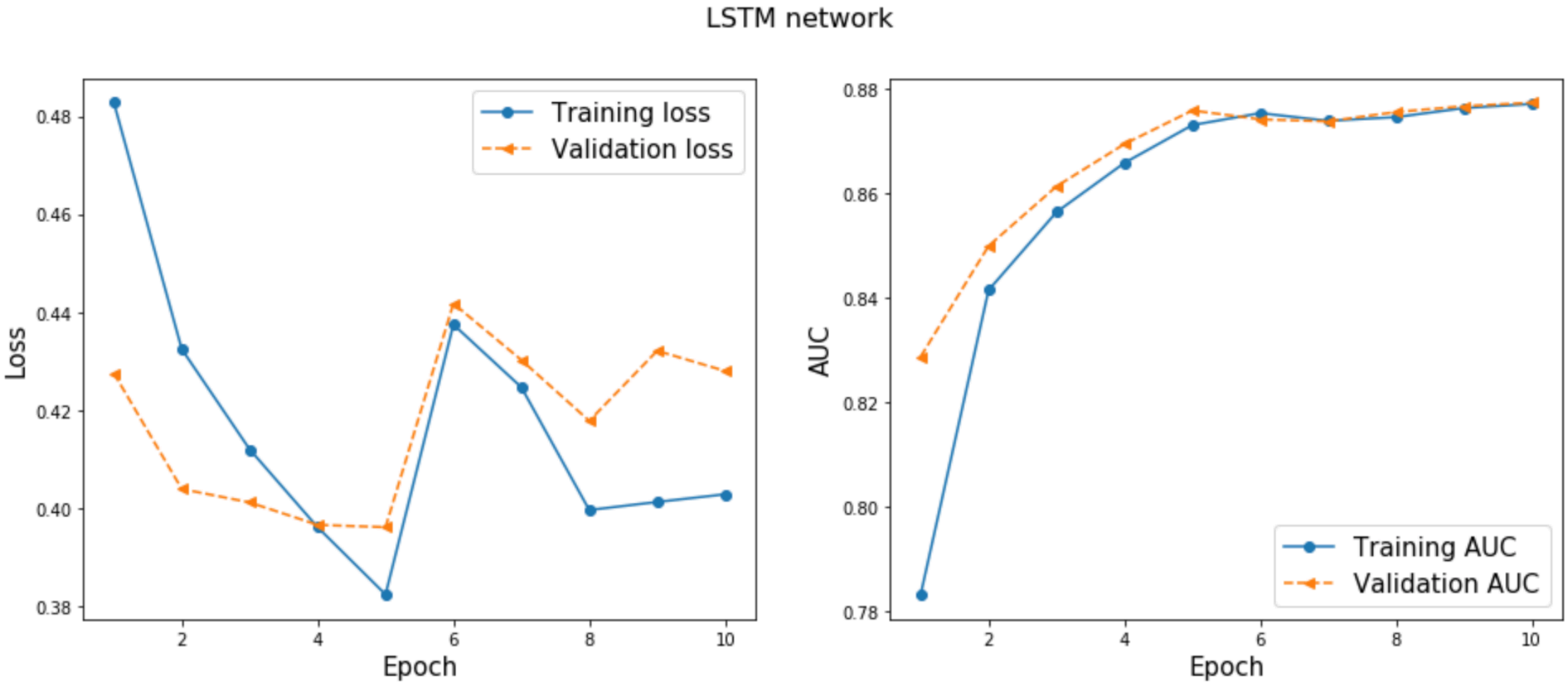
One-shot learning using a twin neural network with with shared LSTM and Manhattan similarity.

- Add LSTM layers, because it is a twin network, both inputs (questions) share the same LSTM.

- Combine the LSTM layers output using the Manhattan distance metric and add it to precalculated similarity features.

*** Manhattan distance:**
The distance between two points measured along axes at right angles. In a plane with p1 at (x1, y1) and p2 at (x2, y2), it is |x1 - x2| + |y1 - y2|.

Network with LSTM - training



Model comparison

Time distributed	precision	recall	f1-score
not similar	0.82	0.84	0.83
similar	0.72	0.68	0.70
max validation AUC			0.85

LSTM	precision	recall	f1-score
not similar	0.81	0.87	0.84
similar	0.75	0.65	0.69
max validation AUC			0.88

Confusion matrix
(time distributed network):

	not duplicate (predicted)	duplicate (predicted)
not duplicate (true)	21410	3972
duplicate (true)	4804	10139

Confusion matrix
(LSTM network):

	not duplicate (predicted)	duplicate (predicted)
not duplicate (true)	22111	3271
duplicate (true)	5278	9665

Conclusions

- The twin neural networks with time distributed layers and LSTM layers were applied on the Quora Question Pairs dataset.
- Jaccard, cosine and Manhattan similarity metrics explored.
- Training the LSTM network was more computationally expensive, compared to the time distributed network, and resulted in moderate increase of precision by 0.03 for duplicate questions (~6 hours for LSTM on google colab notebook with GPU for 10 epochs, compared to ~30 min for time distributed).

***Precision can be interpreted as: if I take a pair of questions predicted as duplicate, what is the probability that it is indeed a duplicate pair of questions.**